

Physics-informed Guided Disentanglement in Generative Networks

Fabio Pizzati, Pietro Cerri, and Raoul de Charette

Abstract—Image-to-image translation (i2i) networks suffer from entanglement effects in presence of physics-related phenomena in target domain (such as occlusions, fog, etc), lowering altogether the translation quality, controllability and variability. In this paper, we propose a general framework to disentangle visual traits in target images. Primarily, we build upon collection of simple physics models, guiding the disentanglement with a physical model that renders some of the target traits, and learning the remaining ones. Because physics allows explicit and interpretable outputs, our physical models (optimally regressed on target) allows generating unseen scenarios in a controllable manner. Secondarily, we show the versatility of our framework to neural-guided disentanglement where a generative network is used in place of a physical model in case the latter is not directly accessible. Altogether, we introduce three strategies of disentanglement being guided from either a fully differentiable physics model, a (partially) non-differentiable physics model, or a neural network. The results show our disentanglement strategies dramatically increase performances qualitatively and quantitatively in several challenging scenarios for image translation.

Index Terms—image to image translation, feature disentanglement, adversarial learning, adverse weather, physics-based rendering, vision and rain, GAN, robotics, autonomous driving, representation learning

1 INTRODUCTION

Image-to-image (i2i) translation GANs can learn complex style mappings in an unsupervised manner, otherwise impractical with traditional physics based rendering. Hence, i2i GANs find great applicability in artistic style transfer, content generation, and other scenarios [1], [2], [3]. When coupled with domain adaptation strategies [4], [5], [6], they also provide an alternative to manual labeling work for synthetic to real [7] or challenging conditions generation [8], [9], [10]. However, a common pitfall of GANs is their inability to accurately learn the underlying physics of the transformation [11], often resulting in artifacts based on inaccurate mapping of source and target characteristics, which significantly impact results. This is the case for example when learning clear \rightarrow rain, as a naive GAN translation will inevitably entangle inaccurate raindrops, as highlighted in Fig. 1 top. On the other hand, physics-inspired models can render well-studied elements of target domain with great realism [12], [13], [14], [15], though leaving any other appearance trait unmodified. For instance, in a rainy scene, models can accurately render raindrops but fail to render the complex scene wetness. We propose a learning-based comprehensive framework to unify generative networks and physics priors. We rely on a disentanglement strategy that benefits from simple physical models to learn the remaining un-modeled mapping. In brief, we render some of the target visual traits with a physical model and learn the un-modeled target characteristics with an i2i network. At inference, we compose them as shown in Fig. 1 to get the output benefiting from the visually pleasant outputs of GANs and the controllable characteristics of physical models. The peculiarity of our method is that we achieve disentanglement of modeled and learned characteristics *by just using data in which*

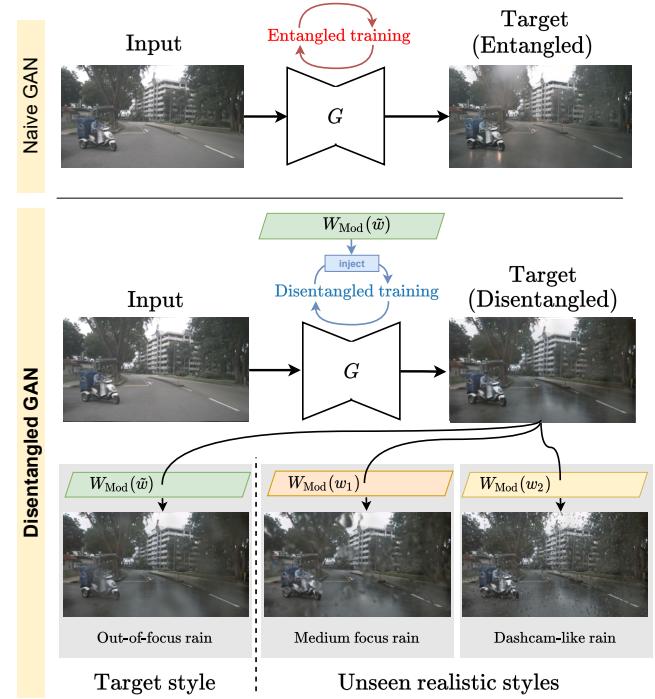


Fig. 1: Guided disentanglement. While naive GANs generate all target scene traits at once (Target - Entangled), we learn a *disentangled* version of the scene from guidance of physical model $W_{Mod}(\cdot)$ with estimated physical parameters (\tilde{w}). Our idea is to combine physical models of well-known phenomena (as raindrops) with generative capabilities of GANs, in a complementary manner. We combine a physical model for raindrops with wetness learned by the GAN (Target - Disentangled), *by only training on entangled data (i.e. rainy scene with raindrops on the lens)*. See the unrealistic raindrops in naive GANs. We instead enable the generation of target style (\tilde{w}) or unseen scenarios (here, w_1, w_2).

- *Fabio Pizzati and Raoul de Charette are with Inria (France)
E-mail: fabio.pizzati@inria.fr, raoul.de-charette@inria.fr*
- *Fabio Pizzati and Pietro Cerri are with Vislab Ambarella (Italy)
Email: c-fpizzati@ambarella.com, pcerri@ambarella.com*

they are both present simultaneously. For example, we can learn to generate wet scenes *without* raindrops on the lens, by only looking at rainy images *with* raindrops. Our strategy deeply differs from sequential composition of i2i and physics based rendering [15] which instead assume underlying independence of the two. Besides increasing image realism, our physical model-guided framework enables fine-grained control of physical parameters in rendered scenes, for increasing generated images variability regardless of the training dataset. This is beneficial for robotics applications, which require resistance to unobserved scenarios. A remarkable use case is vision in rainy conditions since raindrops appearances vary drastically with the camera setup. From Fig. 1 bottom, our disentanglement can be used to be resistant to dashcam-like rain even having only seen out-of-focus rain at training. Other applications we demonstrate in this paper are: vision for dirty images, foggy weather, or composite watermarks.

This research greatly extends our prior work [16] that focused only on occlusion disentanglement for differentiable models. We propose novel contributions that aim to address a wider spectrum of disentanglement cases, and tackle scenarios in which differentiable physical models are cumbersome to use or unavailable. In practice, we extend our model-guided strategy to non-differentiable models (Sec. 3.4), new geometry-dependent task ('Fog' in Sec. 5.1.2), expanding the evaluation qualitatively and quantitatively (Sec. 5.2.1). We also conducted an extensive user study to increase the reliability of our evaluation (Secs. 5.1.5, 5.2.1, 5.3). We extend our general framework to the neural-guided disentanglement setting (Sec. 4), with new ad-hoc experiments (Sec. 5.2.2). We also extend altogether our adversarial parameters estimation (Sec. 5.3), ablations (Sec. 5.4) and discussion (Sec. 6). Finally, to encourage research in this direction, we release the code to replicate our results: <https://github.com/astra-vision/GuidedDisent>.

2 RELATED WORKS

2.1 Image-to-image translation

The seminal work on image-to-image translation (i2i) using conditional GANs on paired images was conducted by Isola et al. [3], while [17] exploits multi-scale architectures to generate HD results. Zhu et al. [1] propose a framework working with unpaired images introducing cycle consistency, exploited also in early work on paired multimodal image translation [18]. A similar idea is proposed in [19].

There has been a recent trend for alternatives to cycle consistency for appearance preservation in several approaches [20], [21], [22], to increase focus on global image appearance and reduce it on unneeded textural preservation. In [23], they propose a cycle consistency-free multi-modal framework. Many methods also include additional priors to increase translation consistency, using objects [9], [24], instance [25], geometry [26], [27], [28] or semantics [29], [30], [31], [32], [33], [34], [35], [36], [37]. Other approaches learn a shared latent space using a Variational Autoencoder, as in Liu et al. [2].

Recently, attention-based methods were proposed, to modify partly input images while keeping domain-invariant regions unaltered [38], [39], [40], [41], [42]. Alternatively, spatial attention was exploited to drive better the adversarial training on unrealistic regions [43]. Some methods focus instead on generating intermediate representations of source and target [44], [45] or continuous translations [46], [47]. In the recent [48], authors exploit similarity with retrieved images to increase translation quality.

2.2 Disentanglement in i2i

Disentangled representations of content and appearance seem to be an emerging trend to increase i2i outputs quality. Recently, Park et al. [49] proposed a contrastive learning based framework to disentangle content from appearance based on patches. MUNIT [50], DRIT [51] and TSIT [52] exploit disentanglement between content and style to achieve one-to-many translations. The idea is further extended in FUNT [53], COCO-FUNIT [54] and ManiFest [55] to achieve few-shot learning, and in TUNIT [56] to translate without source/target distinctions. In HiDT [57], they exploit multi-scale style injection to reach translations of high definition, while [58], [59] conditions disentanglement on domain supervision. Following different reasoning, [60] disentangles representations enforcing orthogonality. In [61], they prevent semantic entanglement by using gradient regularization.

Multi-domain i2i methods [62], [63], [64], [65], [66], [67], [68] could be also exploited for disentangling representations among different domains, at the cost of requiring annotated datasets with separated physical characteristics – practically inapplicable for real images. Recent frameworks [69], [70] unify multi-domain and multi-target i2i exploiting multiple disentangled representations. Some works [71], [72] detach from literature proposing hierarchical generation. In [7], instead, they learn separately albedo and shading, regardless of the general scene. A similar result is performed by [73], only using unpaired images. Recently, VAE-based alternatives have also emerged [74].

Disentangled representations could also help in physics-informed i2i tasks, such as [75] where a fog model is exploited to dehaze images. Similarly, Gong et al. [76] perform fog generation exploiting paired simulated data. Even though these methods effectively learn physical transformations in a disentangled manner, they simply ignore the mapping of other domain traits.

2.3 Physics-based generation

Many works in literature rely on rendering to generate physics-based traits in images, for rain streaks [14], [15], [77], [78], [79], snow [80], fog [14], [81] or others. In many cases, physical phenomena cause occlusion of the scene – well studied in the literature. For instance, many models for raindrops are available, exploiting surface modeling and ray tracing [12], [82], [83]. In [84], raindrop motion dynamics are also modeled. Recent works instead focus on photorealism relaxing physical accuracy constraints [13], [85]. A general model for lens occluders has been proposed in [86]. Logically, it is extremely challenging to entirely simulate the appearance of scene encompassing multiple physical phenomena (for rain: rain streaks, raindrops on the lens, reflections, etc.), hence in [15], [87] they also combine i2i networks and physics-based rendering. In [88], they propose to exploit night physics characteristics to perform domain adaptation. However, this is quite different from our objective since they assume to physically model features not present in the target images. To the best of our knowledge, there is no method which unifies rendering based on physical models and i2i translations in a complementary manner.

3 PHYSICAL MODEL-GUIDED DISENTANGLEMENT

Standard i2i GANs solely rely on context mapping between source and target only – which would be impractical relying only on physical rendering. In some setups, however, the target domain encompasses some visual traits, for example adverse weather or

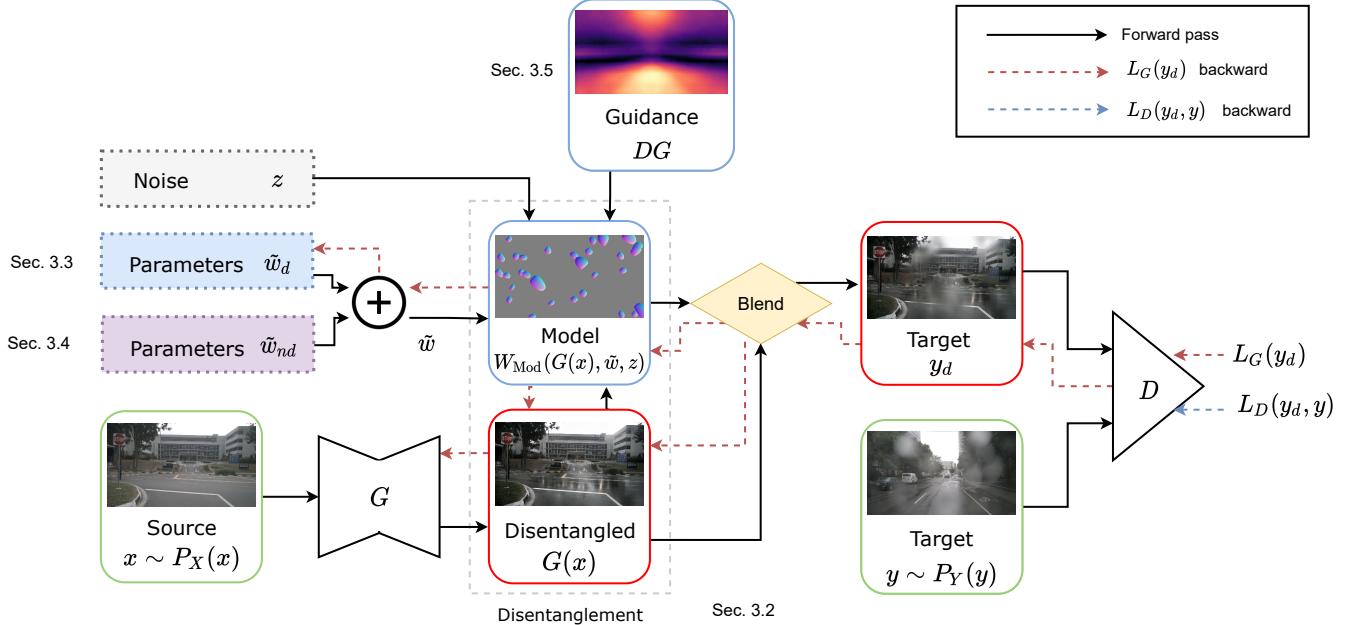


Fig. 2: Model-guided disentanglement. Our unsupervised disentanglement process consists of applying a physical model $W_{\text{Mod}}(\cdot)$ to the generated image $G(x)$. Subsequently, the composite image is forwarded to the discriminator and the GAN loss (L_G or L_D) is backpropagated (dashed arrows). The model rendering depends on the estimated parameters \tilde{w} , composed by differentiable (\tilde{w}_d) and non-differentiable ones (\tilde{w}_{nd}). We use a Disentanglement Guidance (DG) to avoid interfering with the gradient propagation in the learning process. Green stands for real data, red for fake ones.

lens occlusions, whose modeling is well understood from physics. Hence, it may be amenable to integrate *a priori* physics knowledge in the adversarial learning process.

To formalize i2i transformations as a composition of physics and learned characteristics, we propose a setting shown in Fig. 2 where the GAN learns to disentangle the physically modeled traits from target (Sec. 3.1). Disentanglement is achieved relying on physical model-guided strategies (Sec. 3.2), where we exploit as the only prior the nature of the physical trait we aim to disentangle (e.g. raindrop, dirt, fog, etc.). Because these may have infinite variations of appearances, we estimate differentiable (Sec. 3.3) and non-differentiable (Sec. 3.4) target parameters of the physical model which ease disentanglement by reducing differences with target. Our approach boosts image quality and realism guiding model injection during training with gradient-based guidance (Sec. 3.5). An extensive explanation of training strategies is in Sec. 5.

3.1 Adversarial disentanglement

In image-to-image translation we aim to learn a transformation between a source X and a target Y , thus mapping $X \mapsto Y$ in an unsupervised manner. We assume that Y appearance is partly characterized by a well-identified phenomenon such as occlusions on the lens (e.g. rain, dirt) or weather phenomena (e.g. fog). Hence, we propose a sub-domain decomposition (as in [10]) of $Y = \{Y_W, Y_T\}$, separating the identified traits (Y_W) from the other ones (Y_T). We assume this only on target, so $X = \{X_T\}$. In adversarial learning, the task of the generator is to approximate the probability distributions P_X and P_Y associated with the problem domains, such as

$$\begin{aligned} \forall x \in X, x &\sim P_X(x), \\ \forall y \in Y, y &\sim P_Y(y). \end{aligned} \quad (1)$$

For explaining the intuition, we assume that the traits identifiable in this manner are independent from the recorded scene. For instance, physical properties of raindrops on a lens (such as thickness or position) do not change with the scene, as it happens also with fog, where visual effects are only depth-dependent. Therefore, Y_W is fairly independent from Y_T , hence we formalize P_Y as a joint probability distribution with independent marginals, such as

$$P_Y(y) = P_{Y_W, Y_T}(y_W, y_T) = P_{Y_W}(y_W)P_{Y_T}(y_T). \quad (2)$$

Intuitively, approximating one of the marginals with *a priori* knowledge will force the GAN to learn the other one in a disentangled manner. During training, this translates into injecting features belonging to Y_W before forwarding the images to the discriminator, which will provide feedback on the general realism of the image.

Formally, we modify a LSGAN [90] training, which enforces adversarial learning minimizing

$$\begin{aligned} y_d &= G(x), \\ L_{\text{gen}} &= L_G(y_d) = \mathbb{E}_{x \sim P_X(x)}[(D(y_d) - 1)^2], \\ L_{\text{disc}} &= L_D(y_d, y) = \mathbb{E}_{x \sim P_X(x)}[D(y_d)^2] + \\ &\quad + \mathbb{E}_{y \sim P_Y(y)}[(D(y) - 1)^2], \end{aligned} \quad (3)$$

where L_{gen} and L_{disc} are tasks of generator G and discriminator D , respectively. We instead learn a disentangled mapping injecting physically modeled traits $W_{\text{Mod}}(\cdot)$ on translated images. We newly define y_d as the disentangled composition of translated scene $G(x)$ and $W_{\text{Mod}}(\cdot)$, hence

$$y_d = \alpha_w G(x) + (1 - \alpha_w) W_{\text{Mod}}(\cdot). \quad (4)$$

We define as α_w a pixel-wise measure of blending between modeled

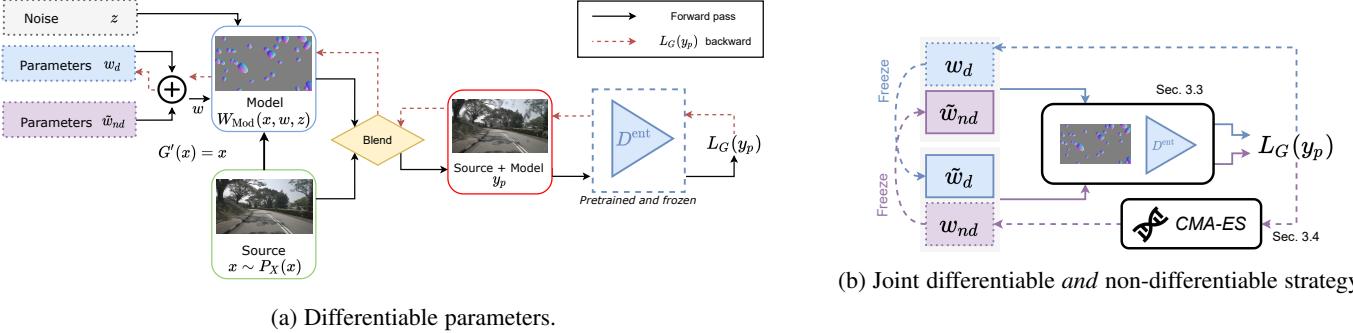


Fig. 3: Model-guided parameters estimation. **a)** We exploit a pretrained discriminator D^{ent} , to calculate an adversarial loss L_G on *source* data augmented with the model W_{Mod} having differentiable parameters w_d . In this process, the gradient flows only in direction of the differentiable parameters. **b)** We optimize until convergence differentiable (blue) and non-differentiable (purple) parameters, alternatively reaching new minima (\tilde{w}_d and \tilde{w}_{nd}) used during optimization of the other parameter set. While differentiable parameters are regressed (Sec. 3.3), non-differentiable ones require black-box genetic optimization (Sec. 3.4), here CMA-ES [89].

and learned scene traits. Pixels which depend only on $W_{\text{Mod}}(\cdot)$ (as opaque occlusions) will show $\alpha_w = 1$ while others (e.g. transparent ones) will have $\alpha_w < 1$.

3.2 Physics models as guidance

One can easily obtain physical model (i.e. W_{Mod}) from existing literature – typically to render visual traits like drops, fog, or else. Injecting such physical models in our guided-GAN enables disentanglement and learning of visual traits *not* rendered by physical models, like wet materials for rain models [14], clouds in the sky for fog models [81], etc.

However, these models often have extremely variable appearance depending on their physical parameters w so we propose adversarial-based strategies to regress optimal \tilde{w} mimicking the target dataset appearance. This is in fact needed for disentangled training where we assume modeled traits to resemble target ones. Other parameters are of stochastic nature (e.g. drop positions on the image) and are encoded as noise z regulating random characteristics. Additionally, some models appearance – like refractive occlusions – varies with the underlying scene¹ s , so we write $W_{\text{Mod}}(\cdot) = W_{\text{Mod}}(s, w, z)$, with $s = G(x)$. Following our pipeline in Fig. 2, if \tilde{w} properly estimates *target* physical parameters, $W_{\text{Mod}}(s, \tilde{w}, z)$ estimates marginal $P_{Y_W}(y_W)$ which again enables disentanglement.

During inference instead, w and z can be arbitrarily varied, greatly increasing generation variability while still obtaining a realistic target scene rendering. In the following, we describe our adversarial parameter estimation strategy, while distinguishing differentiable (w_d) and non-differentiable (w_{nd}) parameters, such that $w = \{w_d, w_{nd}\}$.

3.3 Differentiable parameters estimation

To estimate the target optimized derivable parameters \tilde{w}_d , we exploit an adversarial-based strategy benefiting from entanglement in naive trainings. We consider a naive baseline trained on source \mapsto target mapping, where target entangles two sub-domains as specified in Sec. 3.1. We refer to generator and discriminator trained in this way as *entangled* generator and discriminator, respectively. The entangled discriminator D^{ent} successfully learns

to distinguish fake target images. This results in being able to discriminate $P_X = P_{X_T}$ from $P_Y = P_{Y_T}(y_T)P_{Y_W}(y_W)$. Considering a simplified scenario where P_{Y_T} is arbitrarily confused with the source domain, such that $P_{Y_T} = P_{X_T}$, regressing w_d is the only way to minimize the domain shift. In other words, considering the derivable model parametrized by w_d , the above domain confusion prevents any changes in the scene. To minimize differences between source and target the network is left with updating the injected physical model appearance, ultimately regressing w_d . Fig. 3a shows our differentiable parameter pipeline. From a training perspective, we first pretrain an i2i baseline (e.g. MUNIT [50]), learning a $X \mapsto Y$ mapping with an entangled generator G^{ent} and discriminator D^{ent} . We then freeze D^{ent} and use it to solve

$$y_p = \alpha_w x + (1 - \alpha_w)W_{\text{Mod}}(x, w, z), \min_{w_d} L_G(y_p), \quad (5)$$

backpropagating the GAN loss through the differentiable model. Since many models may encompass pixelwise transparency, often the blending mask α_w is $\alpha_w = \alpha_w(w, z)$. Please note this is *not* a traditional adversarial training, since freezing the discriminator is mandatory to preserve the previously learned target domain appearance during the estimation process. After convergence, we extract the optimal parameter set \tilde{w}_d . Alternatively, \tilde{w}_d could be manually tuned by an operator, at the cost of menial work and inaccuracy, possibly leading to errors in the disentanglement.

From Fig. 3a, notice that the gradient flows only through differentiable parameters (w_d). We now detail our strategy to optimize jointly inevitable non-differentiable parameters (w_{nd}).

3.4 Non-differentiable parameters estimation

The previously described strategy only holds for differentiable parameters w_d , since we use backpropagation of an adversarial loss. Nonetheless, many models include non-differentiable parameters w_{nd} that could equally impact the realism of our model $W_{\text{Mod}}(\cdot)$. For example, a model generating raindrops occlusion would include differentiable parameters like the imaging focus, but also non-differentiable ones like the shape or number of drops – all of which significantly impact visual appearance. However, the sizing of non-differentiable parameters w_{nd} is both complex and time-consuming (as evaluated in Sec. 5.4), and incorrect sizing is likely to achieve suboptimal disentanglement. Manual approximation of optimal w_{nd} parameters via trial-and-

1. In Sec. 6, we explain how W_{Mod} depending of s is not violating the independence assumption of Eq. 2, and evaluate its effect in Sec. 5.4.

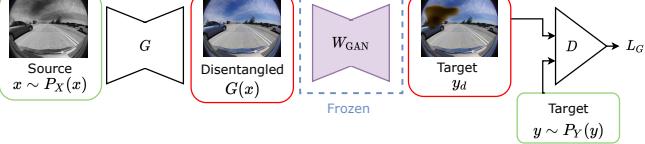


Fig. 4: Neural-guided disentanglement. We exploit here a separate frozen GAN (W_{GAN}) which renders specific target traits (here, dirt) on generator G output images before forwarding them to the discriminator D . We do not show gradient propagation for simplicity.

error might also be cumbersome or impractical for vast search space. To circumvent this, we exploit a genetic strategy estimating w_{nd} .

In our method, non-differentiable parameters are fed to a genetic optimization strategy. The evolutionary criteria remain the same as for differentiable parameters, that is the pretrained discriminator (D^{ent}) adversarial loss. In practice, to avoid noisy updates after genetic estimation, we average adversarial loss over a fixed number of samples to reliably select a new population. After convergence, we extract the optimal parameter set \tilde{w}_{nd} . In our experiments, we use CMA-ES [89] as evolutionary strategy, but the proposed pipeline is extensible to any other genetic algorithm.

3.5 Disentanglement guidance

It is worth noting that too sparse injection of model $W_{\text{Mod}}(\cdot)$ negatively impacts disentanglement because the guided-GAN will entangle similar physical traits to fool the discriminator, while injecting too much of $W_{\text{Mod}}(\cdot)$ will prevent the discovery of the disentangled target. Spatially, we observe that regions that do not differ from source to target are most frequently impacted by entanglement. This is because the discriminator naturally provides less reliable predictions due to the local source-target similarities, which leads the generator to produce artifacts resembling target physical characteristics to fool the discriminator, eventually leading to unwanted entanglement. In rainy scenes this happens for trees or buildings, whose appearance little varies if dry or wet, whereas ground or road exhibit puddles which are strong rainy cues.

To balance the injection of $W_{\text{Mod}}(\cdot)$, we guide disentanglement by injecting $W_{\text{Mod}}(\cdot)$ only on low domain shift areas, pushing the guided-GAN to learn the disentangled mapping of the scene. Specifically, we learn a Disentanglement Guidance (DG) dataset-wise by averaging the GradCAM [91] feedback on the source dataset, relying on the discriminator D^{ent} gradient on *fake* classification. Areas with high domain shift will be easily identified as *fake*, while others will impact less on the prediction. To take into account different resolutions, we evaluate GradCAM for all the discriminator layers. Formally, we use LSGAN to obtain

$$\text{DG} = \mathbb{E}_{x \sim P_X(x)} [\mathbb{E}_{l \in L} [\text{GradCAM}_l(D^{\text{ent}}(x))]], \quad (6)$$

with L being the discriminator layers. At training, we inject models only on pixels (u, v) where $\text{DG}_{u,v} < \gamma$, with $\gamma \in [0, 1]$ as hyperparameter. In Sec. 5.4 we visually assess the effect of DG.

3.6 Training strategy

For models having differentiable *and* non-differentiable parameters we employ a joint optimization shown in Fig. 3b. We first initialize a set of parameters w , then alternatively use our strategy for

differentiable parameters estimation w_d (Sec. 3.3) and the genetic strategy for non differentiable ones w_{nd} (Sec. 3.4). Notice that the alternation of optimized parameters prevents divergence due to simultaneous optimization. We apply updates until optimum, reaching the two sets of target style parameters, $\tilde{w} = \{\tilde{w}_d, \tilde{w}_{nd}\}$. The complete training strategy for model-guided disentanglement is in Sec. 5.1.1.

4 NEURAL-GUIDED DISENTANGLEMENT

For some visual traits, a physical model may not be immediately available so we consider also the case in which the guidance is provided by a neural model, learned separately. Referring to our adversarial strategy in Sec. 3.1, we simply substitute W_{Mod} with W_{GAN} in Eq. 4, where W_{GAN} is our neural guidance – a GAN in our experiments. Following our past explanations, assuming W_{GAN} generates specific visual traits – may it be dirt, drop, watermark or else – it is an approximation of the marginal $P_{Y_W}(y_W)$. We define $\tilde{\theta}$ as the optimal set of parameters of the network to reproduce target occlusion appearance. Subsequently, processing generated images with W_{GAN} before forwarding them to the discriminator pushes the guided-GAN we aim to train in a disentangled manner (not to be confused with W_{GAN}) to achieve disentanglement, as illustrated in Fig. 4, following the same reasoning as in Sec. 3.1.

Of importance here, even if W_{GAN} is trained supervisedly – for example, from annotated pairs of images / dirt – the disentanglement strategy is itself fully unsupervised. Also, referring to Eq. 2, the guided-GAN can only achieve disentanglement and estimate $P_{Y_T}(y_T)$ from images in Y , if W_{GAN} (i.e. $W(\cdot)$) correctly estimates $P_{Y_W}(y_W)$. Suppose W_{GAN} augments rain on images, it will be sensitive to the intensity as well as the appearance of drops of Y . In other words, it would be possible only to recreate target-like scenes, being only able to modify parameters that do not depend from appearance, as raindrops position. With the model-guided disentanglement strategy we could instead re-inject physical traits of arbitrary appearance, greatly increasing the generative capabilities of our guided framework. Hence, the primary goal of this pipeline shall not be seen as a competitor to model-based disentanglement, but rather as a viable alternative when a physical model is not available. We present the training strategy in Sec. 5.1.1.

5 EXPERIMENTS

We evaluate our disentanglement strategies on the real datasets nuScenes [92], RobotCar [85], Cityscapes [93] and WoodScape [94], and on the synthetic Synthia [95] and Weather Cityscapes [14]. Our evaluation methodology is in Sec. 5.1 including training, tasks, user study, and model/neural guidance.

In Sec. 5.2 we extensively study the disentanglement of raindrop, dirt, composite occlusions, and fog – on a qualitative/quantitative basis, and using proxy tasks and human judgement. Our method is compared against the recent DRIT [51], U-GAT-IT [41], AttentionGAN [40], CycleGAN [1], and MUNIT [50] frameworks. Opposite to the literature, our method enables disentanglement of the target domain, so we report both the disentangled translations as well as the translations with the injection of optimal target physical traits. The disentanglement is greatly visible in images presented in this section. Because physical models are readily available, we emphasize our physical model-guided strategy (Sec. 3) evaluated on 4 models in Sec. 5.2.1. Conversely, the neural-guided strategy (Sec. 4), requires rare separate neural networks

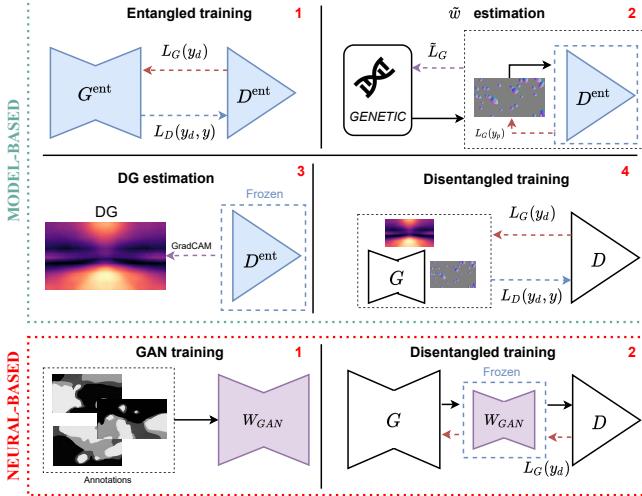


Fig. 5: **Training pipelines.** For *model-guided* disentanglement, we 1) train a naive i2i entangled baseline, 2) use the entangled discriminator feedback to estimate optimal parameters \tilde{w} and 3) Disentanglement Guidance (DG), and finally 4) train the guided-GAN with model injection. For *neural-guided* disentanglement, we 1) train a GAN (W_{GAN}) exploiting additional knowledge as semantics and 2) use it to inject target traits during our guided-GAN training.

for rendering traits. It is subsequently only evaluated on dirt disentanglement in Sec. 5.2.2, relying on DirtyGAN [96], for comparison purposes with the model-guided strategy. In Sec. 5.3, we study the accuracy of our physical model parameters estimation on the well-documented raindrop model, and finally ablate our proposal in Sec. 5.4.

Formalism. We formalize disentangled trainings as \mathcal{T}_{dis} , guided either with a full physical model ($\mathcal{T}_{W_{Mod}}$), a model with only differentiable parameters ($\mathcal{T}_{W_{Mod}^{wd}}$), or neural-guided ($\mathcal{T}_{W_{GAN}}$). When re-injecting physical traits, we also show their parameters in parentheses. For example, $\mathcal{T}_{W_{Mod}}(\tilde{w})$ means model-guided disentangled output with injection of the full model estimated on target (\tilde{w}).

5.1 Methodology

5.1.1 Training

Our disentangled GAN is architecture agnostic. Here, we rely on the MUNIT [50] backbone for its multi-modal capabilities, and exploit LSGAN [90] for training. Fig. 5 shows our two training pipelines.

For **model-guided** training (Fig. 5, top), we leverage on a multi-step pipeline, only assuming the known nature of features to disentangle (e.g. raindrop, dirt, fog, etc.). First, an i2i source \mapsto target baseline is trained in an entangled manner, obtaining entangled discriminator (D^{ent}). Second, we make use of D^{ent} to regress the optimal parameters \tilde{w} with adversarial (Sec. 3.3) and genetic (Sec. 3.4) estimation. Third, we extract Disentanglement Guidance (Sec. 3.5), also using D^{ent} . Finally, we train from scratch the disentangled guided-GAN (Sec. 3).

For **neural-guided** training (Fig. 5, bottom), we use a prior-agnostic two-step pipeline. First, we train the third-party W_{GAN} to render occlusions, exploiting semantic supervision in our experiments though it could realistically be replaced with self-supervision. Then, we train our disentangled guided-GAN *without* any supervision.

	Task	Entanglement	Datasets	Guidance
Model	clear \mapsto rain _{drop}	Raindrop	nuScenes [92]	Raindrop w_d σ $t, (s, p)$
	gray \mapsto color _{dirt}	Dirt	WoodScape [94]	Dirt σ, α
	synth \mapsto WCS _{fog}	Fog	Synthia [95], Weather CS [14]	Fog β
	clear \mapsto snow _{cmp}	Composite	Synthia [95]	Composite
Neural	gray \mapsto color _{dirt}	Dirt	WoodScape [94]	Network DirtyGAN [96]

TABLE 1: **Disentanglement tasks.** For each task, we indicate the features entangled in the target domain (also, shorten as indices of task name), the datasets, and the model or neural guidance employed for disentanglement.

5.1.2 Tasks

Tab. 1 lists the tasks evaluated and ad-hoc datasets. When referring to a task, we denote as indices the entangled features in target domain. Thus, clear \mapsto rain_{drop} literally means ‘translation from clear to rain with entangled drops’. We later describe models used for disentanglement.

clear \mapsto rain_{drop} We exploit the recent nuScenes [92] which includes urban driving scenes, and use metadata to build clear/rain splits obtaining 114251/29463 training and 25798/5637 testing clear/rain images. Target rain images entangle highly unfocused drops on the windshield, which would hardly be annotated as seen in Fig. 6, first row.

gray \mapsto color_{dirt} Here, we rely on the recent fish-eye WoodScape [94] dataset which has some images with soiling on the lens. We separate the dataset in clean/dirty images using soiling metadata getting 5117/4873 training images and 500/500 for validation. Because clean/dirty splits do not encompass other domain shifts, we additionally transform *clean* images to *gray*. Subsequently, we frame this as a colorization task where target *color* domain entangles *dirt*. For disentanglement, we experiment using both a physical model-guided and a neural-guided strategy.

clear \mapsto snow_{cmp} With Synthia [95] we also investigate entanglement of very different alpha-blended composites, like “Confidential” watermarks or fences. We split Synthia using metadata into clear/snow images and further augment snow target with said composite at random position. As clear/fog splits, we use 3634/3739 images for training and 901/947 for validation. To guide disentanglement, we consider a composite model, inspiring from the concept of thin occluders [77].

synth \mapsto WCS_{fog} We learn here the mapping from synthetic Synthia [95] to the foggy version of Weather CityScapes [14] – a foggy-augmented Cityscapes [93]. The goal is to learn the synthetic to real mapping, while disentangling the complex fog effect in target. For training we use 3634/11900 and 901/2000 for validation as Synthia/WeatherCityscapes. We use a fog model to guide our network.

Note that this task differentiates from others, since target has fog of heterogeneous intensities (max. visibility 750, 375, 150 and 75m) making disentanglement significantly harder.

5.1.3 Physical model guidance

To correctly fool the discriminator, it is crucial to choose a model that realistically resembles the entangled feature. We

leverage 4 physical models, listed in Tab. 1 ‘Model’ with their differentiable (w_d) and non-differentiable (w_{nd}) parameters.

Raindrop model. We extend the model of Alletto *et al.* [13], which is balanced between complexity and realism. Drops are approximated by simple trigonometric functions, while we encompass also noise addition for shape variability [97]. For drops photometry, we use fixed displacement maps (U, V) for coordinate mapping on both x and y axes, technically encoded as 3-channels images [13]. To approximate light refraction, a drop at (u, v) has its pixel (u_i, v_i) mapped to

$$(u + U(u_i, v_i) \cdot \rho, v + V(u_i, v_i) \cdot \rho), \quad (7)$$

where ρ is a drop-wise value representing water thickness. Most importantly, we also model imaging focus, since it may extremely impact the rendered raindrop appearance [13], [98], [99]. Hence, we use a Gaussian point spread function [100] to blur synthetic raindrops. We implement kernel variance σ as differentiable, while drops size (s), frequency (p), and shape (t) related parameters are non differentiable. We use a single shape parameter and generate 4 types of drops, with associated p and t .

Dirt model. Here, we naively extend our raindrop model removing displacement maps as soil has no refractive behaviors. Instead, we introduce a color guidance that forces synthetic dirt to be brighter in peripherals regions, also depending on a parameter α which regulates occlusion maximum opacity (hence, maximum α_w value). We also estimate σ as aforementioned.

Composite occlusions model. We exploit the model of thin occluder proposed in [77] to render composite occlusions on images, i.e. randomly translated alpha-blended transparent images such as watermarks or fence-like grids. We assume to fully know transparency, thus no parameter is learned.

Fog model. We leverage the physics model of [14] using an input depth map. Fog thickness is regulated by a differentiable extinction coefficient β which regulates maximum visibility.

5.1.4 Neural guidance

Finding appropriate neural networks to render visual traits is not trivial. Here we experiment only with Dirt, as listed in Tab. 1 ‘Neural’.

Dirt neural. DirtyGAN [96] is a GAN-based framework for opaque soiling occlusion generation. It is composed by two components, i.e. a VAE for occlusion map generation (trained using soiling semantic maps) and an i2i network conditioned on the generated map to include synthetic soiling on images. To train DirtyGAN, we first train a VAE to learn the shape of soiling, and then proceed to train a modified CycleGAN [1] to generate realistic soiling, conditioning the soiling shape on the VAE outputs. For more details on this we refer to [96].

5.1.5 User study

We also conducted a qualitative anonymous online study collecting answers from 56 users (22 males, 33 females, 1 non-binary) from 21 to 65 years old (mean 27.9, std. 7.6). Each user had to evaluate 85 randomized scenes with a Likert-5 scale, providing the image looks realistic and efficiently disentangled. For ease of reading we included the results in each ad-hoc subsections (Secs. 5.2.1, 5.3).

5.2 Disentanglement

In this section, we evaluate our disentanglement strategy both using physical model-guidance (Sec. 5.2.1) or neural-guidance (Sec. 5.2.2).

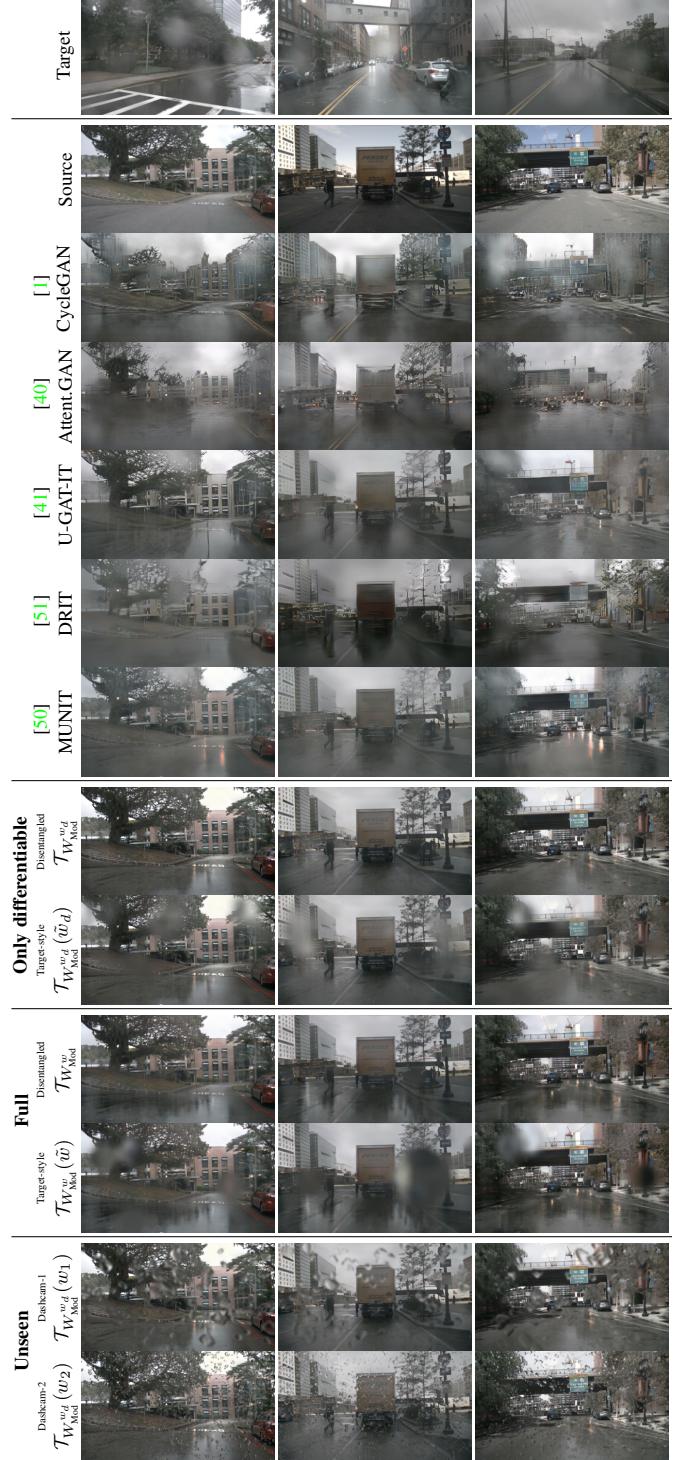


Fig. 6: **Raindrop disentanglement on clear \rightarrow rain_{drop}.** We compare qualitatively with the state-of-the-art on the clear \rightarrow rain_{drop} task with rain drops model-guided disentanglement. In the first row, we report samples of the target domain. Subsequently, the *Source* image (2nd row), the translations by different baselines (rows 3-7) and our results (rows 8-13). Our model-guided network is able to disentangle the generation of peculiar rainy characteristics from the drops on the windshield ('Disentangled' rows) and re-injection with estimated parameters ('Target-style'). We evaluate both the differentiable-only parameter estimation (rows 8-9) and the genetic-based full estimation (rows 10-11). We also show injection of other arbitrary parameters w_1, w_2 (last 2 rows).

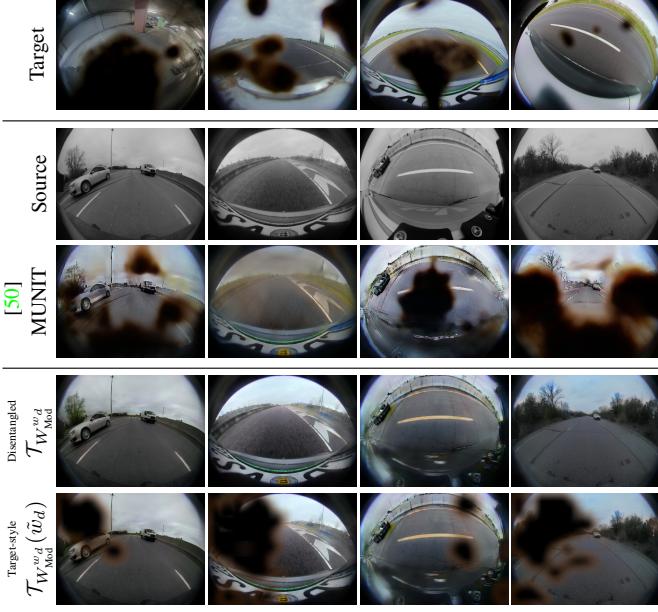


Fig. 7: **Dirt disentanglement on gray \mapsto color_{dirt}**. We compare with MUNIT [50] for the gray \mapsto color_{dirt} task. Although MUNIT successfully mimics the *Target* style (rows 1,3), our approach lead to a more realistic image colorization disentangling the presence of dirt ('Disentangled' row $\mathcal{T}_{W_{Mod}}^{w_d}$). We also use the dirt model to reproduce Target images ('Target-style' row $\mathcal{T}_{W_{Mod}}^{w_d}(\tilde{w})$).

5.2.1 Physical model-guided

Referring to the 4 tasks and 4 ad-hoc models in Tab. 1 ‘Model’, we evaluate our ability to disentangle visual traits with our physical model guidance from Sec. 3, reporting quality, quantitative and human judgment.

Hereafter, we separate experiments on Raindrop, Dirt and Composite disentanglement from the Fog experiments, since only the former have homogeneous physical parameters (w) throughout the dataset². Since non-differentiable parameters were fairly easy to manually tune, we thoroughly experiment in the differentiable-only $\{w_d\}$ setup and compare it later on to our full $\{w_d, w_{nd}\}$ estimation (Sec. 5.3).

Qualitative disentanglement. We present different outputs for the clear \mapsto raindrop trained on nuScenes [92], comparing to state-of-the-art methods [1], [40], [41], [50], [51] (Fig. 6) and for gray \mapsto color_{dirt} and clear \mapsto snow_{cmp} with respect to the backbone (Figs. 7,8, respectively). In all cases, baselines entangle occlusions in different manners. For instance, in Fig. 6 it is noticeable the constant position of rendered raindrops between different frameworks, as in the 1st column on the leftmost tree, which is a visible effect of entanglement and limits image variability. Also, occlusion entanglement could cause very unrealistic outputs where the structural consistency of either the scene (Fig. 7) or the occlusion (Fig. 8) is completely lost.

Referring to Figs. 6,7,8, our method is always able to produce high quality images *without* occlusions ('Disentangled' rows) including typical target domain traits such as wet appearance without drops, colored image without dirt or snowy image without occlusions, respectively. Furthermore, we can inject occlusions

2. For *Raindrop*, *Dirt* and *Composite* we consider w_d and w_{nd} to be dataset-wise constant. E.g. all raindrops have the same defocus blur, transparency, etc. Conversely, *Fog* images have varying fog intensity.

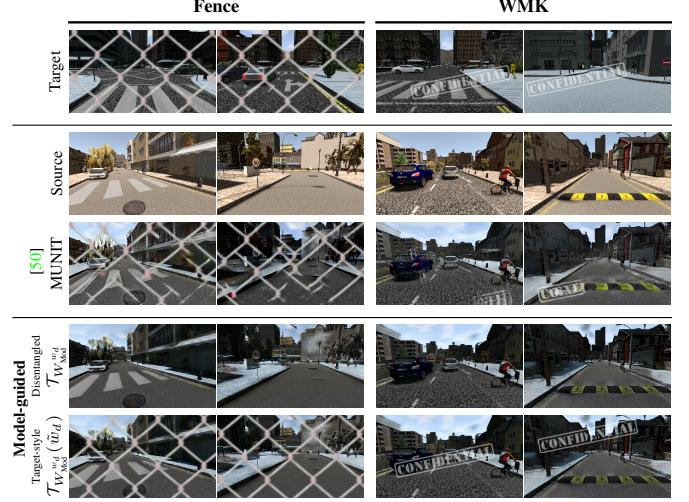


Fig. 8: **Composite disentanglement on clear \mapsto snow_{cmp}**. We extend the applicability of our method to composite occlusions, that we validate in the clear \mapsto snow_{cmp} scenario. We add a fence-like occlusion (left) and a *confidential* watermark (right) to *synthetic_snow*, with random position. As expected, we encounter entanglement phenomena for MUNIT, while our model-guided network is successful in learning the disentangled appearance ('Disentangled' row $\mathcal{T}_{W_{Mod}}^{w_d}$). In our 'Target-style' row $\mathcal{T}_{W_{Mod}}^{w_d}(\tilde{w}_d)$, we inject the occlusions to mimic the target style.

with optimal estimated parameters ('Target-style' rows) to mimic target appearance which enables a fair comparison with baselines³.

We also inject raindrops with arbitrary parameters to simulate *unseen* dashcam-style images in Fig. 6 (last 2 rows). The realistic results demonstrate both the quality of our disentanglement and the realism of the *Raindrop* model.

Quantitative disentanglement. We use GAN metrics to quantify the quality of the learned mappings. Results are reported in Tab. 2a, where Inception Score (IS) [101] evaluates quality and diversity against target, LPIPS distance [102] evaluates translation diversity (thus avoiding mode-collapse), and Conditional Inception Score (CIS) [50] single-image translations diversity for multi-modal baselines. In practice, IS is computed over all the validation set while CIS is estimated on 100 different translations of 100 random images following [50]. The InceptionV3 network for Inception Scores was finetuned on the source/target classification as in [50]. LPIPS distance is calculated on 1900 random pairs of 100 translations as in [50]. For fairness, we only compare 'Target-style' outputs to baselines, since those are not supposed to disentangle physical traits, and can only output images resembling *Target*. Tab. 2a shows we outperform all baselines on IS/CIS, including MUNIT – our i2i backbone. This is due to disentanglement, since entanglement phenomena limit occlusions appearance and position variability. Even the scene translation quality is improved by disentanglement since the generator learns a simpler target domain mapping without any occlusions. As regards LPIPS distance, we outperform the baseline on raindrops while we rank lower on the other tasks. While IS/CIS quantify both quality and diversity, LPIPS metric is evaluating variability only thus penalizing simpler occlusion generation. For instance, our rendered dirt in Fig. 7 is often black while MUNIT-generated artifacts are highly variable

3. For comparing with neural methods we set $\alpha = 1$ (cf. Sec. 5.2.2).

Experiment	Network	IS↑	LPIPS↑	CIS↑
clear \mapsto rain _{drop}	CycleGAN [1]	1.15	0.473	-
	AttentionGAN [40]	1.41	0.464	-
	U-GAT-IT [41]	1.04	0.489	-
	DRIT [51]	1.19	0.492	1.12
	MUNIT [50]	1.21	0.495	1.03
	Ours $\mathcal{T}_{W_{\text{Mod}}}^w(\tilde{w})$	1.25	0.502	1.08
	Ours $\mathcal{T}_{W_{\text{Mod}}^{wd}}(\tilde{w}_d)$	1.53	0.515	1.15
gray \mapsto color _{dirt}	MUNIT [50]	1.06	0.656	1.08
	Ours $\mathcal{T}_{W_{\text{Mod}}^{wd}}(\tilde{w}_d)$	1.25	0.590	1.15
clear \mapsto snow _{cmp} (fence)	MUNIT [50]	1.26	0.547	1.11
	Ours $\mathcal{T}_{W_{\text{Mod}}^{wd}}(\tilde{w}_d)$	1.31	0.539	1.19
clear \mapsto snow _{cmp} (WMK)	MUNIT [50]	1.17	0.567	1.01
	Ours $\mathcal{T}_{W_{\text{Mod}}^{wd}}(\tilde{w}_d)$	1.19	0.551	1.02
synth \mapsto WCS _{fog}	CycleGAN [1]	1.31	0.384	-
	AttentionGAN [40]	*	*	*
	U-GAT-IT [41]	1.05	0.406	-
	DRIT [51]	1.22	0.424	1.10
	MUNIT [50]	1.22	0.429	1.13
	Ours $\mathcal{T}_{W_{\text{Mod}}^{wd}}(\tilde{w}_d)$	1.33	0.420	1.17

* AttentionGAN converges to the identity transformation.

(a) GAN metrics.

Method	AP↑
Original (from [14])	18.7
Finetuned w/ Halder <i>et al.</i> [14]	25.6
Finetuned w/ Model-guided $\mathcal{T}_{W_{\text{Mod}}^{wd}}(\tilde{w}_d)$	27.7

(b) Semantic segmentation on rain.

TABLE 2: **Image quality evaluation.** In (a), we quantify GAN metrics for all tasks. While quality-aware metrics are always successfully increased, LPIPS depends on the visual complexity of the model and presence of artifacts. In (b), we compare our pipeline for finetuning semantic segmentation network outperforming the state-of-the-art for rain generation.

(compare rows *MUNIT* and ours $\mathcal{T}_{W_{\text{Mod}}^{wd}}(\tilde{w}_d)$). The same happens for watermarks in Fig. 8, where unrealistic artifacts are highly variable. For raindrops, instead, *MUNIT* tends to just blur images, while we benefit from the refractive capabilities of our physical model which increase LPIPS.

Semantic segmentation. To provide additional insights on the effectiveness of our framework and compensate for the well-known noisiness of GAN metrics [102], we quantify the usability of generated images for semantic segmentation in the clear \mapsto rain_{drop} setup. Therefore, we process the popular Cityscapes [93] dataset for semantic segmentation with our best clear \mapsto rain_{drop} model-guided training, obtaining a synthetic rainy version $\mathcal{T}_{W_{\text{Mod}}^{wd}}(\tilde{w}_d)$ that we use for finetuning PSPNet [103], following Halder *et al.* [14]. Please note that this also demonstrates the generation capabilities to new scenarios of our GAN, since we use the pretrained network on nuScenes given the absence of rainy scenes in Cityscapes. We report the mAP for the 25 rainy images with semantic labels provided by [14] in Tab. 2b. We experience a significant increase (+9%) with respect to baseline PSPNet trained on original clear images (*Original*), and also outperform (+2.1%) the finetuning with rain physics-based rendering [14]. Both networks finetune *Original* weights. The overall low numbers reported are impacted by the significant domain shift between Cityscapes and nuScenes.

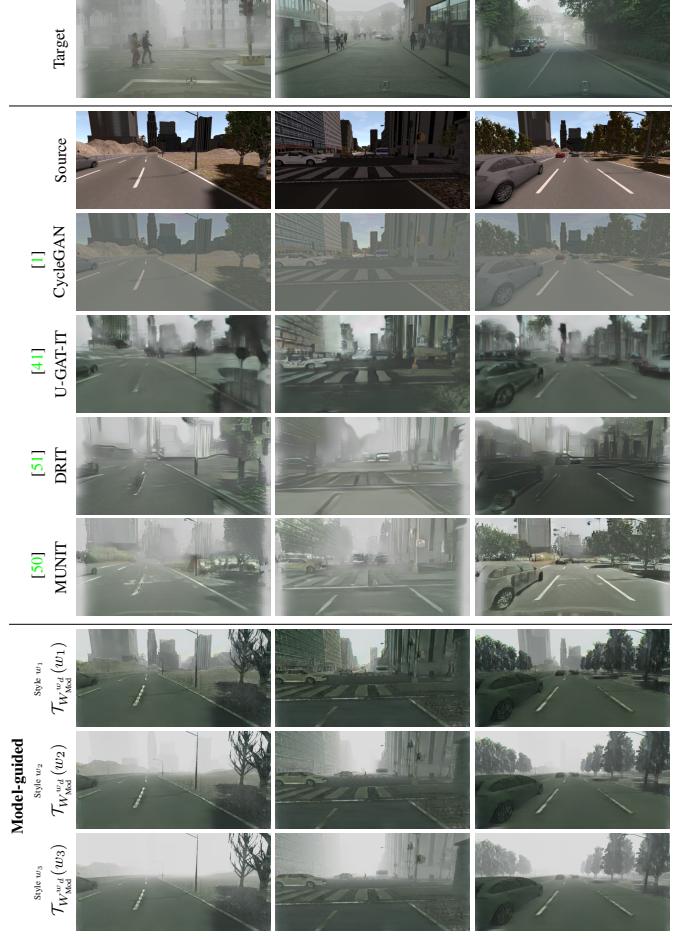


Fig. 9: **synth \mapsto WCS_{fog} translations.** As visible, MUNIT shows entanglement phenomena, leading to artifacts. Our model-guided disentanglement, instead, enables to generate a wide range of foggy images, with arbitrary visibility, while maintaining realism. Since the fog model W_{Mod} always blocks the gradient propagation in the sky region, the network can not achieve photorealistic disentanglement but still improves the generated image quality.

Disentanglement on heterogeneous datasets. We now evaluate the effectiveness of the synth \mapsto WCS_{fog} experiment which translates from synthetic Synthia to the real-augmented Weather CityScapes [14] entangling fog of various intensities (from light to thick fog). Notice this task significantly differs from others for two reasons. First, unlike other experiments the model parameter – the optical extinction coefficient, β – varies in the target dataset. Second, the fog model is depending on the scene geometry [104]. This makes the disentanglement task non-trivial. In our adversarial disentanglement, we however still regress a single $\beta = 28.61$ somehow averaging the ground truth values ($\beta \in [4, 40]$).

In Fig. 9 results show we are able to generate images stylistically similar to target ones, but with geometrical consistency and varying β (last 3 rows). Instead, MUNIT [50] fails to preserve realism due to entanglement artifacts, visible in particular on elements at far (as buildings in the background). Please note that we intentionally do not show disentangled output for fairness, since the physical model always blocks the gradient propagation in the sky. More details on this will be discussed in Sec. 6. Randomizing $\beta \in [4, 40]$ we report GAN metrics results in Tab. 2a, where the increased quality of images is quantified. LPIPS distance suffers

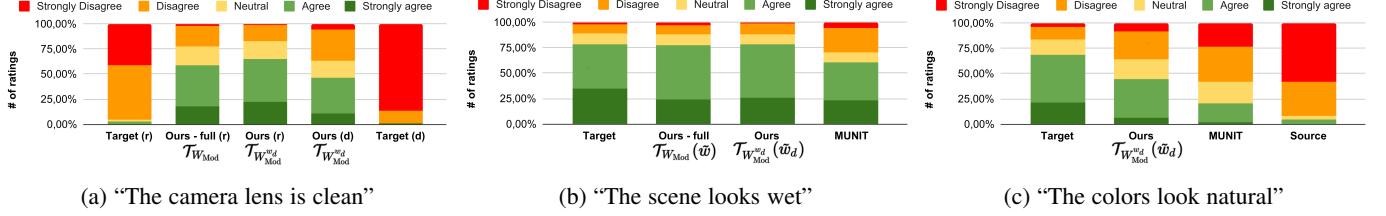


Fig. 10: Disentanglement user study. We asked 56 users (cf. Sec. 5.1.5) to judge the lens cleanliness (a) on raindrops (r) and dirt (d), or the wetness (b) or coloring (c) of clear \mapsto rain_{drop} and gray \mapsto color_{dirt} generated scenes, respectively. Details are in the text. Our system greatly improves results following human evaluation metrics.

from the absence of artifacts in our model-guided $T_{W_{Mod}^{w_d}}(\tilde{w}_d)$, which artificially increases image variability. The physical model always renders correctly regions at far (e.g. the sky, which is always occluded), hence pure variability quantified by LPIPS is reduced (cf. above LPIPS definition).

User study. To further evaluate our disentanglement quality, we asked 56 users to rate images (details in Sec. 5.1.5). First, we presented our disentangled outputs and real images *with* occlusions on the clear \rightarrow rain_{drop} and gray \rightarrow color_{dirt} tasks, where users were asked to rate for each image if "The camera lens is clean (no dirt, no raindrops)". Results in Fig. 10a show our strategy is better since the lens in our images is judge cleaner than target images. However, this does not assess if the underlying transformation (i.e. wetness or color) was properly learned.

Hence, secondly we compare translation realism with the MUNIT baseline, rating the statement "The scene looks wet" for clear \rightarrow rain_{drop} and "The scene looks colorful" for gray \rightarrow color_{dirt}. We also include real source images (i.e. gray) in gray \rightarrow color_{dirt} to evaluate performances in the naive identity transformation, and target images in both to set upper bounds. Results in Figs. 10b, 10c clearly show the superiority of our approach with respect to the MUNIT, heavily reducing the gap with real target images.

In a nutshell, the study demonstrates that disentanglement is fairly perceived by users (Fig. 10a) while preserving the learned underlying transformation (Figs. 10b, 10c).

5.2.2 Neural-guided disentanglement

Referring to Tab. 1 ‘Neural’, we now evaluate our ability to disentangle visual traits with our neural guidance from Sec. 4, for Dirt disentanglement in the gray \mapsto color_{dirt} task, by using available annotations instead of a physics-based prior.

Evaluation. We leverage here the WoodScape [94] datasets having soiling semantic annotation as polygons. Following our training strategy (Fig. 5, bottom), our neural guidance DirtyGAN [96] (cf. Sec. 5.1.4) is trained beforehand and frozen during the disentanglement.

The use of annotations boosts the overall quality and diversity, which is proved in Tab. 11a where our neural-guided outperforms both MUNIT baseline and our own model-guided version. Furthermore, since the ground truth for colorization is available, we evaluate in Tab. 11b the effectiveness of disentanglement with SSIM and PSNR metrics (higher is better). Here both disentanglement outperform MUNIT [50] significantly, but model-guided is better. Arguably, we attribute this to the worse gradient propagation due

to more occluded pixels with respect to our physical model⁴.

Finally, last 2 rows of Fig. 11c show our neural-guided strategy produces high quality *colored* images *without* occlusions (‘Disentangled’ row, $T_{W_{GAN}}$) while injection of occlusions with optimal estimated parameters $\tilde{\theta}$ (‘Target-style’ row, $T_{W_{GAN}}(\tilde{\theta})$) also mimics target appearance. In fact while both neural-guided disentanglement (Sec. 5.2.2) and physical model-guided disentanglement (Sec. 5.2.1) perform well, only our model-guided strategies controllability of the occlusion at inference. This is because of the explicit physical parameters in the models, that allows reinjecting unseen models at inference.

5.3 Parameters estimation

We now evaluate the effectiveness of our parameter estimation for physical model-guided disentanglement, considering only differentiable parameters first and later extending to our full system. The neural-guided disentanglement strategy precludes this analysis due to the lack of explicit parameters.

Differentiable model ($w = \{w_d\}$). To evaluate realism, we leverage the RobotCar [85] dataset having pairs of clear/raindrop images. Since there is no domain shift between image pairs, we set $G(x) = x$ and regress the defocus blur (σ) again following Sec. 3.3. The regressed $\sigma = 3.87$ is used to render raindrops on clear images. Using FID and LPIPS distances we measure perceived distance between real raindrop images and our model-guided raindrops translations ($T_{W_{Mod}^{w_d}}(\tilde{w}_d)$) or the one of Porav et al. [85]. Fig. 12b shows we greatly boost similarity⁵ (-72.02 FID) with real raindrop images. This is qualitatively verified in Fig. 12a, where our rendered raindrops are more similar to *Target*. To provide insights about the quality of our minima, we also evaluate FID for arbitrary σ values ($\sigma \in \{0.0, 2.5, 5.0, 7.5, 10\}$). Fig. 12c proves that our estimated sigma best minimized perceptual distances despite the weak discriminator signal.

To measure the accuracy of our differentiable parameter regression (Sec. 3.3) we need paired images with and without physical traits with completely known physical parameters. To the best of our knowledge such dataset does not exists. Instead, we augment RobotCar [85], WoodScape [94] and Synthia [95] with synthetic raindrops, dirt, and fog, respectively, with gradually increasing values of defocus blur (σ) for raindrop, transparency (α) for dirt⁶ and optical thickness (β) for fog. Using each augmented

4. On average, DirtyGAN dirt covers 25.4% of the image while our physical model covered 20.1%. While this provides more realistic dirt masks (ground truth annotation is 29.6%) we conjecture this leads to worse gradient propagation.

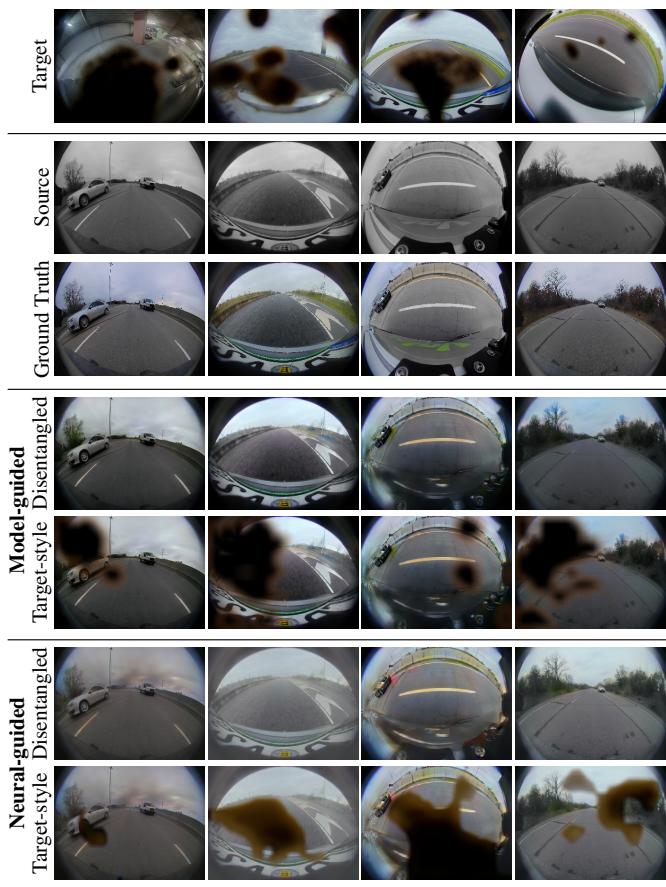
5. Please note that unlike previous experiments, here LPIPS is used for distance estimation (not diversity), so lower is better.

6. In this experiment, we consider dirt with a fixed defocus blur value σ and regress only α to increase the diversity of tasks.

Network	IS↑	LPIPS↑	CIS↑	Network	SSIM↑	PSNR↑
MUNIT [50]	1.06	0.656	1.08	MUNIT [50]	0.414	13.4
Model-guided $\mathcal{T}_{W_{\text{Mod}}}^{w_d}(\tilde{w}_d)$	1.25	0.590	1.15	Model-guided $\mathcal{T}_{W_{\text{Mod}}}^{w_d}$	0.755	20.2
Neural-guided $\mathcal{T}_{W_{\text{GAN}}}(\tilde{\theta})$	1.58	0.663	1.47	Neural-guided $\mathcal{T}_{W_{\text{GAN}}}$	0.724	19.3

(a) GAN metrics.

(b) Colorization.



(c) Qualitative evaluation.

Fig. 11: **Comparison of model- and neural- guided disentanglement on gray \mapsto color_{dirt}.** Although our neural-guided strategy excels in image quality and diversity, mostly due to the complex nature of generated dirt (a), with model guidance we achieve more realistic image colorization (b). Qualitative results are coherent with metrics (c). With both pipelines, we still outperform MUNIT [50], used as backbone.

dataset, we then regress said parameters following Sec. 3.3.

Plots in Fig. 13 show estimation versus ground-truth. In average, the estimation error is 0.99% for raindrop, 3.55% for dirt, and 23.51% for fog. The very low σ error for raindrop is to be imputed to the defocus blur that drastically changes scene appearance, while higher error for β must be imputed to the logarithmic dependency of the fog model. Nevertheless, translations preserve realism (cf. Fig. 9).

Full model ($w = \{w_d, w_{nd}\}$). To evaluate the quality of our full raindrop model, we incorporate this time the non-differentiable parameters (i.e. s, p, t) which are estimated with our genetic strategy in Sec. 3.4 for 4 types of drops, with a genetic population size of 10. As shown in Fig. 12b, LPIPS metric privileges our full model-

guided estimation ($\mathcal{T}_{W_{\text{Mod}}}(\tilde{w})$) while FID suffers compared to using differentiable parameters only. However, we very significantly outperform [85] also qualitatively (Fig. 12a). The mitigated results are explained by the much more complex optimization problem having many more parameters, and by the limited computation time for genetic iterations. However, this let us foresee applications in high-dimensionality problems where manual approximation is not always possible or with a less accurate model (see ablations Sec. 5.4). Moreover, we stress that the manual optimization could be challenging and time consuming (cf. Sec. 5.4). Results on the clear \mapsto rain_{drop} task in Fig. 14 are coherent with above insights as the full model estimation, although effective, exhibits slightly lower quality disentanglement.

User study. We presented to users (see Sec. 5.1.5) couples of images with independent scenes in which the left one presented images with real drops taken from RobotCar [85], while the right one included fake drops rendered with our model with differentiable only / all parameters estimated, or with Porav et al. [85]. Users were asked to compare raindrops appearance between the two images regardless of the represented scenes. From results shown in Fig. 15, it is evident that our method largely outperform the baseline in both configurations, indicating a higher quality of our raindrops also for the human preference metric.

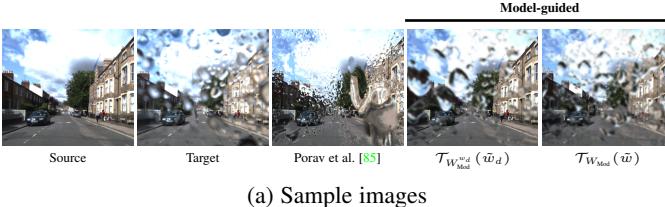
5.4 Ablation studies

We now ablate our proposal. We focus on the model-guided setting by tuning genetic processing, altering model complexity, changing models, or removing disentanglement guidance. We also further investigate and compare different training strategies.

Model complexity. We study the influence of the model on disentanglement for the clear \mapsto rain_{drop} on nuScenes [92] task. Specifically, we evaluate three raindrop models of decreasing complexity: 1) Our model from Sec. 5.1.3 (named *Ours*). 2) The same model but without shape and thickness variability (*Refract*), and 3) A naive non-parametric colored Gaussian-shape model (*Gaussian*). Note that *Gaussian* is deprived of any refractive property as it uses fixed color, and does not regress any physical parameters. In Fig. 16a, we report GAN metrics for all models following Sec. 5.2.1. Even if increasing model complexity is beneficial for disentanglement, very simple models still lead to a performance boost. We advocate the best performances of *Ours* to a more effective discriminator fooling during training, as consequence of increased realism.

Model choice. To also evaluate whether injected features only behave as adversarial noise regardless of the chosen model, we trained on RobotCar [85] (as in Sec. 5.3) though purposely using an incorrect model as watermark, dirt, fence. Evaluating the FID against real raindrop images, we measure 135.32 (raindrop) / 329.17 (watermark) / 334.76 (dirt) / 948.71 (fence), proving necessity of using the ad-hoc model.

Disentanglement Guidance (DG). We use the nuScenes clear \mapsto rain_{drop} task to visualize the effects of different DG strategies (Sec. 3.5). For varying values of the DG threshold γ in Fig. 16b we see results ranging from no guidance ($\gamma = 0$) to strict guidance ($\gamma = 1$). With lax guidance ($\gamma = 0$), we fall back in the baseline scenario with visible entanglement effects, while with $\gamma = 1$ we do achieve disentanglement, at the cost of losing important visual features as reflections on the road. Only appropriate guidance ($\gamma = 0.75$) achieves disentanglement and preserves realism.



Method	FID_{\downarrow}	$\text{LPIPS}_{\downarrow}$
Porav et al. [85]	207.34	0.53
Model-guided $T_{W_{Mod}^{wd}}(\bar{w}_d)$	135.32	0.44
Model-guided $T_{W_{Mod}}(\bar{w})$	157.44	0.43

(b) Benchmark on [85]

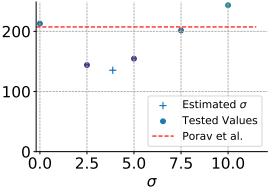


Fig. 12: **Realism of the injected occlusion.** Our defocus blur σ estimation grants an increased realism in raindrop rendering on the RobotCar [85] dataset (a), compared with Porav et al. [85]. This is confirmed by quantitative metrics (b). We report our model-guided translations using either differentiable parameter estimation only ($T_{W_{Mod}^{wd}}(\bar{w}_d)$) or the full model parameter estimation ($T_{W_{Mod}}(\bar{w})$), outperforming Porav et al. [85] in both. In (c), we evaluate the FID for different σ values in [0, 10], showing that our regressed σ value ($\sigma = 3.81$) actually leads to a local minimum.

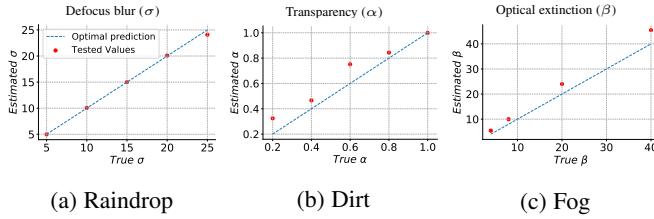


Fig. 13: **Evaluation of the model parameters regression.** The reliability of our parameter estimation is assessed on synthetic datasets augmented with arbitrary physical models acting as ground truth values. Comparing against our regressed value, our strategy performs better when low modifications on the estimated values corresponds to big visual changes (average error is 0.99% for raindrops (a), 3.55% for dirt (b)). For fog (c), we get an higher error of 23.51% due to the low visual impact of high β values.

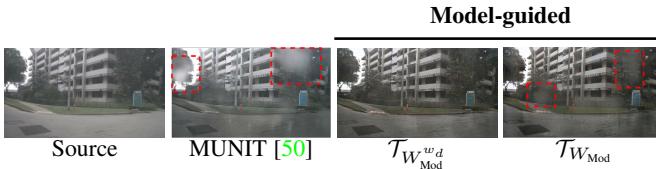


Fig. 14: **Full model on clear \leftrightarrow raindrop.** With complete parameter estimation ($T_{W_{Mod}}$, rightmost), we achieve a slightly worse disentanglement than with manually-tuned non-differentiable parameters ($T_{W_{Mod}^{wd}}$), visible in red areas of $T_{W_{Mod}}$. However, in both of our translations we generate typical rain traits as reflections with reasonable disentanglement, while baseline MUNIT [50] has very evident raindrops entangled highlighted in red.

5.4.1 Full model

Non-differentiable genetic estimation. We study the effectiveness of our genetic estimation ablating the population size of our raindrop model on RobotCar [85] as in Sec. 5.3. We test our algorithm with population size 10/25/50/100, obtaining FID 157.44/153.32/151.21/**149.09** and LPIPS **0.43**/0.44/0.44/**0.43**. While we observe an obvious increase in performances, this comes with additional computation times, hence we used the lowest population size of 10 for all tests. Nevertheless, this opens doors to potential improvements in the full parametric estimation.
Non-differentiable boundaries of w_{nd} . Genetic algorithms requires optimization boundaries for each parameter (i.e. the *min* and the *max* of each parameter), so one could argue that

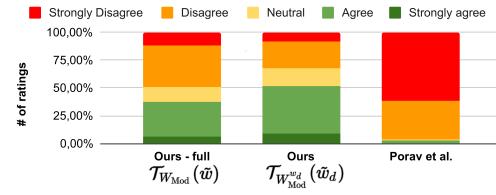


Fig. 15: **Parameter estimation user study.** We presented users with {Reference, Model} image pairs where Reference includes real drops and Model has fake drops rendered with our method with differentiable only (Ours) or full (Ours - full) parameters estimation, or with Porav et al. [85]. Users were asked whether they agree on the statement "The drops of the Model resemble the drops of Reference". Thanks to our estimation strategy, we dramatically improve similarity to real raindrops.



Fig. 16: **Ablations of model complexity and Disentanglement Guidance.** In (a), we quantify disentanglement effects with simpler model having less variability (*Refract*), or only color guidance (*Gaussian*). Even if complexity is beneficial for disentanglement (*Ours*), simple models permits disentanglement to some extent. In (b), we study the efficacy of the Disentanglement Guidance (DG) for different γ values on clear \leftrightarrow raindrop task. With $\gamma = 0$ our approach fallbacks to the baseline and entangles occlusions, while with guidance $\gamma = 1$ the translation lacks important features such as reflections and glares. With $\gamma = 0.75$ we simultaneously avoid entanglements and preserve translation capabilities.

w_{nd} still requires manual tuning, therefore lowering the interest of our full estimation pipeline. However, our empirical studies demonstrate that parameter boundaries only takes a few minutes, while precise manual tuning required for differentiable-only optimization (Sec. 3.3) takes days as it requires multiple training. In an effort to provide evidence of the coarse boundaries definition, we randomly sampled 5 sets of w_{nd} within said boundaries



Fig. 17: Benefit of genetic parameters optimization. While genetic algorithms require to set optimization boundaries, even coarsely-defined boundaries can be used for achieving disentanglement. We sample 5 different random sets of parameters ($\{w_{nd}^{\text{rand-1}}, \dots, w_{nd}^{\text{rand-5}}\}$) from boundaries set in minutes and combine them with \bar{w}_d estimation, achieving visible entanglement artifacts (highlighted with red boxes). Instead, using the same coarsely-defined boundaries for our genetic optimization our full parameter estimation $\{\bar{w}, \bar{w}_{nd}\}$ achieves reasonable disentanglement and qualitatively better results. Hence, our full optimization pipeline can benefit even from quick and coarse tuning of parameter boundaries.

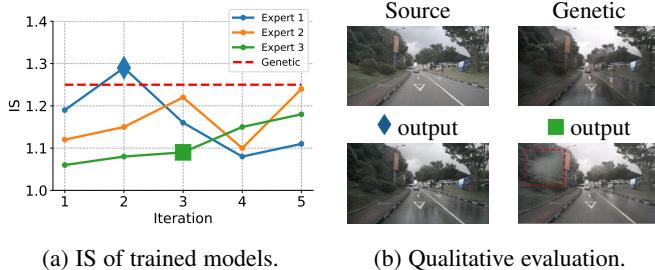


Fig. 18: User-based VS Genetic-based optimization of w_{nd} . In (a), we compare IS of user- or genetic-optimized w_{nd} . Expert computer vision users struggle to reach performances comparable to our genetic estimation due to the complexity of the parameter estimation task. In (b), qualitative evaluation of the GAN output advocates that manual tuning can still lead to good performances (♦ model), but it can also lead to entanglement even after several iterations (■ model). Tuning w_{nd} with our full pipeline (Genetic) prevents such failures, requiring also only one disentangled training.

and report disentanglement results in Fig. 17. Simply sampling parameters within the boundaries (center) achieve far less good disentanglement w.r.t. our full estimation pipeline (right).

5.4.2 Differentiable-only model

Manual estimation of w_{nd} . To provide further proof on the interest of optimizing w_{nd} with genetic algorithms, we perform an additional user study with three computer vision experts. To each expert, we show real rainy images of clear \mapsto rain_{drop} (see Tab. 1) and ask the latter to manually tune w_{nd} of the drop model to reproduce the target drop appearance. We then estimate the optimal remaining differentiable parameters \bar{w}_d and train a disentangled network, showing to the same expert the qualitative results obtained with the tuned parameters. We finally asked to update the manually estimated values to improve disentanglement. We perform multiple iterations and quantify performances in terms of Inception Score (IS). In Fig. 18, it is visible how users difficultly improve performances even after 5 iterations, while with the full estimation pipeline we boost results with no manual tuning. Altogether, this demonstrates how using our full pipeline can ease the estimation task and save computational time.

6 DISCUSSION

To our best knowledge, we have designed the first unsupervised strategy to disentangle physics-based features in i2i. Good qualitative and quantitative performances showcase promising interest for

Guidance	Param. estimation	Editable	Requirements			
			Annotations	Ad-hoc GAN	Model design	Manual w_{nd} tuning
Neural Model	None w_d	✓	✓	✓		
Model	{ w_d, w_{nd} }	✓			✓	✓

TABLE 3: Comparison of the disentanglement strategies. Model-guided strategies do not require annotations and ad-hoc generative networks, but they rely on the availability of a somehow realistic physics model. When using neural-guided disentanglement, the ability to modify physical parameters of the model (“Editable”) is lost. We overcome the need of cumbersome manual tuning of w_{nd} with genetic optimization in our full strategy. However, results in Sec. 5 advocate that best disentanglement performances are still obtained by manually sizing each non differentiable parameter, at the cost of intensive labor and many trainings.

several applications, still there are peculiar points and limitations which we now discuss.

Comparison of different disentanglement strategies. We propose three different disentanglement strategies. In Tab. 3, we compare them, highlighting advantages and disadvantages that could be crucial for choosing a disentanglement strategies in an applicative scenario. While the differentiable-only estimation strategy performs best in terms of disentanglement, it is also time consuming due to manual tuning of w_{nd} . The applicability of neural-guided disentanglement depends on annotations availability, and prevents outputs editing capabilities at inference. Ultimately, one should prefer model-guided disentanglement if a model is accessible.

Independence assumption. For unsupervised disentanglement, we assume the physical model to be completely independent from the scene, in order to use our intuition about marginal separation (see Sec. 3.1 and Eq. 2). However, since physical models may need the underlying scene to correctly render desired traits, one may argue their appearance is not completely disentangled. While this is true from a visual point of view, it is not from a physical one. Let’s interpret disentanglement properties to be dependent on scene *elements*. In presence of disentanglement, the same physical model could be applied to different objects regardless of what they are. For instance, we could use the same raindrop refraction map on either roads or buildings with identical parameters. In this sense, $G(x)$ dependency in physical models is not impacting our visual independence assumption.

On partial entanglement issues. We observe in some cases that gradient propagation can be affected by fixed entanglement of occlusion features. This is the case for example for sky regions in fog (Sec. 5.2.1) because physics [104] formalizes that regardless of its intensity fog is always entangled at far. In such scenarios, disentanglement will perform poorly because the

generator will not get any discriminative feedback. In many other cases however, Disentanglement Guidance (DG, Sec. 3.5) mitigates the phenomenon as it blocks injection of the physical model in relevant image regions. We conjecture that the effectiveness could be extended by varying DG at training time to ensure a balanced gradient propagation.

On genetic estimation effectiveness. The sub-optimal performances of our genetic estimation of w_{nd} are imputed to the much more complex search space, in which we vary all parameters of our physical model simultaneously. It is worth noting that manually tuning non-differentiable parameters requires many trainings, while relying on genetic optimization achieves acceptable results in a single complete training. Also, we did set fairly large search boundaries for w_{nd} (as evaluated in Sec. 5.4), but one could envisage a mixed training in which the search space is limited to reasonable hand-tuned boundaries. In this sense, genetic estimation of w_{nd} could be seen as a minimum mining technique, ensuring increased performances on the hand-tuned values.

Acknowledgments. This work used HPC resources from GENCI-IDRIS (Grants 2020-AD011012040 and 2021-AD011012808). This work was partially funded by French project SIGHT (ANR-20-CE23-0016).

REFERENCES

- [1] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *CVPR*, 2017. [1](#), [2](#), [5](#), [7](#), [8](#), [9](#)
- [2] M.-Y. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-image translation networks,” in *NeurIPS*, 2017. [1](#), [2](#)
- [3] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *CVPR*, 2017. [1](#), [2](#)
- [4] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, E. A. A., and T. Darrell, “CyCADA: Cycle consistent adversarial domain adaptation,” in *ICML*, 2018. [1](#)
- [5] Y. Li, L. Yuan, and N. Vasconcelos, “Bidirectional learning for domain adaptation of semantic segmentation,” in *CVPR*, 2019. [1](#)
- [6] M. Toldo, A. Maracani, U. Michieli, and P. Zanuttigh, “Unsupervised domain adaptation in semantic segmentation: a review,” *Technologies*, 2020. [1](#)
- [7] S. Bi, K. Sunkavalli, F. Perazzi, E. Shechtman, V. G. Kim, and R. Ramamoorthi, “Deep cg2real: Synthetic-to-real translation via image disentanglement,” in *ICCV*, 2019. [1](#), [2](#)
- [8] Y. Qu, Y. Chen, J. Huang, and Y. Xie, “Enhanced pix2pix dehazing network,” in *CVPR*, 2019. [1](#)
- [9] Z. Shen, M. Huang, J. Shi, X. Xue, and T. S. Huang, “Towards instance-level image-to-image translation,” in *CVPR*, 2019. [1](#), [2](#)
- [10] F. Pizzati, R. de Charette, M. Zaccaria, and P. Cerri, “Domain bridge for unpaired image-to-image translation and unsupervised domain adaptation,” in *WACV*, 2020. [1](#), [3](#)
- [11] Y. Xie, E. Franz, M. Chu, and N. Thuerey, “tempoGAN: A temporally coherent, volumetric gan for super-resolution fluid flow,” *SIGGRAPH*, 2018. [1](#)
- [12] M. Roser and A. Geiger, “Video-based raindrop detection for improved image registration,” in *ICCV Workshops*, 2009. [1](#), [2](#)
- [13] S. Alletto, C. Carlin, L. Rigazio, Y. Ishii, and S. Tsukizawa, “Adherent raindrop removal with self-supervised attention maps and spatio-temporal generative adversarial networks,” in *ICCV Workshops*, 2019. [1](#), [2](#), [7](#)
- [14] S. S. Halder, J.-F. Lalonde, and R. de Charette, “Physics-based rendering for improving robustness to rain,” in *ICCV*, 2019. [1](#), [2](#), [4](#), [5](#), [6](#), [7](#), [9](#)
- [15] M. Tremblay, S. S. Halder, R. de Charette, and J.-F. Lalonde, “Rain rendering for evaluating and improving robustness to bad weather,” *IJCV*, 2020. [1](#), [2](#)
- [16] F. Pizzati, P. Cerri, and R. de Charette, “Model-based occlusion disentanglement for image-to-image translation,” in *ECCV*, 2020. [2](#)
- [17] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, “High-resolution image synthesis and semantic manipulation with conditional gans,” in *CVPR*, 2018. [2](#)
- [18] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, “Toward multimodal image-to-image translation,” in *NeurIPS*, 2017. [2](#)
- [19] Z. Yi, H. Zhang, P. Tan, and M. Gong, “Dualgan: Unsupervised dual learning for image-to-image translation,” in *ICCV*, 2017. [2](#)
- [20] M. Amadio and S. Krishnaswamy, “Travelgan: Image-to-image translation by transformation vector learning,” in *CVPR*, 2019. [2](#)
- [21] S. Benain and L. Wolf, “One-sided unsupervised domain mapping,” in *NeurIPS*, 2017. [2](#)
- [22] H. Fu, M. Gong, C. Wang, K. Batmanghelich, K. Zhang, and D. Tao, “Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping,” in *CVPR*, 2019. [2](#)
- [23] O. Nizan and A. Tal, “Breaking the cycle-colleagues are all you need,” in *CVPR*, 2020. [2](#)
- [24] D. Bhattacharjee, S. Kim, G. Vizier, and M. Salzmann, “Dunit: Detection-based unsupervised image-to-image translation,” in *CVPR*, 2020. [2](#)
- [25] S. Mo, M. Cho, and J. Shin, “Instagan: Instance-aware image-to-image translation,” in *ICLR*, 2019. [2](#)
- [26] W. Wu, K. Cao, C. Li, C. Qian, and C. C. Loy, “Transsgaga: Geometry-aware unsupervised image-to-image translation,” in *CVPR*, 2019. [2](#)
- [27] M. Arar, Y. Ginger, D. Danon, A. H. Bermano, and D. Cohen-Or, “Unsupervised multi-modal image registration via geometry preserving image-to-image translation,” in *CVPR*, 2020. [2](#)
- [28] A. Dell’Eva, F. Pizzati, M. Bertozzi, and R. de Charette, “Leveraging local domains for image-to-image translation,” in *VISAPP*, 2022. [2](#)
- [29] P. Li, X. Liang, D. Jia, and E. P. Xing, “Semantic-aware grad-gan for virtual-to-real urban scene adaption,” *BMVC*, 2018. [2](#)
- [30] P. Z. Ramirez, A. Tonioni, and L. Di Stefano, “Exploiting semantics in adversarial training for image-level domain adaptation,” in *IPAS*, 2018. [2](#)
- [31] H. Tang, D. Xu, Y. Yan, J. J. Corso, P. H. Torr, and N. Sebe, “Multi-channel attention selection gans for guided image-to-image translation,” in *CVPR*, 2019. [2](#)
- [32] A. Cherian and A. Sullivan, “Sem-gan: Semantically-consistent image-to-image translation,” in *WACV*, 2019. [2](#)
- [33] Z. Zhu, Z. Xu, A. You, and X. Bai, “Semantically multi-modal image synthesis,” in *CVPR*, 2020. [2](#)
- [34] P. Zhu, R. Abdal, Y. Qin, and P. Wonka, “Sean: Image synthesis with semantic region-adaptive normalization,” in *CVPR*, 2020. [2](#)
- [35] C.-T. Lin, Y.-Y. Wu, P.-H. Hsu, and S.-H. Lai, “Multimodal structure-consistent image-to-image translation,” in *AAAI*, 2020. [2](#)
- [36] L. Ma, X. Jia, S. Georgoulis, T. Tuytelaars, and L. Van Gool, “Exemplar guided unsupervised image-to-image translation with semantic consistency,” in *ICLR*, 2019. [2](#)
- [37] B. Lütjens, B. Leshchinsky, C. Requena-Mesa, F. Chishtie, N. Díaz-Rodríguez, O. Boulaïs, A. Piña, D. Newman, A. Lavin, Y. Gal, and C. Raïssi, “Physics-informed gans for coastal flood visualization,” *arXiv*, 2020. [2](#)
- [38] Y. A. Mejjati, C. Richardt, J. Tompkin, D. Cosker, and K. I. Kim, “Unsupervised attention-guided image-to-image translation,” in *NeurIPS*, 2018. [2](#)
- [39] S. Ma, J. Fu, C. Wen Chen, and T. Mei, “Da-gan: Instance-level image translation by deep attention generative adversarial networks,” in *CVPR*, 2018. [2](#)
- [40] H. Tang, D. Xu, N. Sebe, and Y. Yan, “Attention-guided generative adversarial networks for unsupervised image-to-image translation,” in *International Joint Conference on Neural Networks (IJCNN)*, 2019. [2](#), [5](#), [7](#), [8](#), [9](#)
- [41] J. Kim, M. Kim, H. Kang, and K. Lee, “U-gat-it: unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation,” in *ICLR*, 2020. [2](#), [5](#), [7](#), [8](#), [9](#)
- [42] Y. Lin, Y. Wang, Y. Li, Y. Gao, Z. Wang, and L. Khan, “Attention-based spatial guidance for image-to-image translation,” in *WACV*, 2021. [2](#)
- [43] —, “Attention-based spatial guidance for image-to-image translation,” in *WACV*, 2021. [2](#)
- [44] R. Gong, W. Li, Y. Chen, and L. V. Gool, “Dlow: Domain flow for adaptation and generalization,” in *CVPR*, 2019. [2](#)
- [45] W. Lira, J. Merz, D. Ritchie, D. Cohen-Or, and H. Zhang, “Ganhopper: Multi-hop gan for unsupervised image-to-image translation,” in *ECCV*, 2020. [2](#)
- [46] F. Pizzati, P. Cerri, and R. de Charette, “CoMoGAN: continuous model-guided image-to-image translation,” in *CVPR*, 2021. [2](#)
- [47] Y. Liu, E. Sangineto, Y. Chen, L. Bao, H. Zhang, N. Sebe, B. Lepri, W. Wang, and M. De Nadai, “Smoothing the disentangled latent style space for unsupervised image-to-image translation,” in *CVPR*, 2021. [2](#)
- [48] R. Gomez, Y. Liu, M. De Nadai, D. Karatzas, B. Lepri, and N. Sebe, “Retrieval guided unsupervised multi-domain image to image translation,” in *MM*, 2020. [2](#)
- [49] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, “Contrastive learning for unpaired image-to-image translation,” in *ECCV*, 2020. [2](#)

- [50] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, “Multimodal unsupervised image-to-image translation,” in *ECCV*, 2018. 2, 4, 5, 6, 7, 8, 9, 10, 11, 12
- [51] H.-Y. Lee, H.-Y. Tseng, Q. Mao, J.-B. Huang, Y.-D. Lu, M. Singh, and M.-H. Yang, “Drit++: Diverse image-to-image translation via disentangled representations,” *IICCV*, 2020. 2, 5, 7, 8, 9
- [52] L. Jiang, C. Zhang, M. Huang, C. Liu, J. Shi, and C. C. Loy, “Tsit: A simple and versatile framework for image-to-image translation,” in *ECCV*, 2020. 2
- [53] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz, “Few-shot unsupervised image-to-image translation,” in *ICCV*, 2019. 2
- [54] K. Saito, K. Saenko, and M.-Y. Liu, “Coco-funit: Few-shot unsupervised image translation with a content conditioned style encoder,” in *ECCV*, 2020. 2
- [55] F. Pizzati, J.-F. Lalonde, and R. de Charette, “ManiFest: Manifold Deformation for Few-shot Image Translation,” *arXiv*, 2021. 2
- [56] K. Baek, Y. Choi, Y. Uh, J. Yoo, and H. Shim, “Rethinking the truly unsupervised image-to-image translation,” in *ICCV*, 2021. 2
- [57] I. Anokhin, P. Solovev, D. Korzhenkov, A. Kharlamov, T. Khakhulin, A. Silvestrov, S. Nikolenko, V. Lempitsky, and G. Sterkin, “High-resolution daytime translation without domain labels,” in *CVPR*, 2020. 2
- [58] W. Xia, Y. Yang, and J.-H. Xue, “Unsupervised multi-domain multimodal image-to-image translation with explicit domain-constrained disentanglement,” *Neural Networks*, 2020. 2
- [59] J. Lin, Z. Chen, Y. Xia, S. Liu, T. Qin, and J. Luo, “Exploring explicit domain supervision for latent space disentanglement in unpaired image-to-image translation,” *T-PAMI*, 2019. 2
- [60] R. Kondo, K. Kawano, S. Koide, and T. Kutsuna, “Flow-based image-to-image translation with feature disentanglement,” in *NeurIPS*, 2019. 2
- [61] Z. Jia, B. Yuan, K. Wang, H. Wu, D. Clifford, Z. Yuan, and H. Su, “Lipschitz regularized cyclegan for improving semantic robustness in unpaired image-to-image translation,” *arXiv*, 2020. 2
- [62] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, “Stargan: Unified generative adversarial networks for multi-domain image-to-image translation,” in *CVPR*, 2018. 2
- [63] A. Romero, P. Arbeláez, L. Van Gool, and R. Timofte, “Smit: Stochastic multi-label image-to-image translation,” in *ICCV Workshops*, 2019. 2
- [64] A. Aboosheh, E. Agustsson, R. Timofte, and L. Van Gool, “Combogan: Unrestrained scalability for image domain translation,” in *CVPR Workshops*, 2018. 2
- [65] X. Yang, D. Xie, and X. Wang, “Crossing-domain generative adversarial networks for unsupervised multi-domain image-to-image translation,” in *MM*, 2018. 2
- [66] L. Hui, X. Li, J. Chen, H. He, and J. Yang, “Unsupervised multi-domain image translation with domain-specific encoders/decoders,” in *ICPR*, 2018. 2
- [67] P.-W. Wu, Y.-J. Lin, C.-H. Chang, E. Y. Chang, and S.-W. Liao, “Relgan: Multi-domain image-to-image translation via relative attributes,” in *ICCV*, 2019. 2
- [68] T.-P. Nguyen, S. Lathuilière, and E. Ricci, “Multi-domain image-to-image translation with adaptive inference graph,” in *ICPR*, 2021. 2
- [69] X. Yu, Y. Chen, S. Liu, T. Li, and G. Li, “Multi-mapping image-to-image translation via learning disentanglement,” in *Advances in Neural Information Processing Systems*, 2019. 2
- [70] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, “Stargan v2: Diverse image synthesis for multiple domains,” in *CVPR*, 2020. 2
- [71] K. K. Singh, U. Ojha, and Y. J. Lee, “Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery,” in *CVPR*, 2019. 2
- [72] X. Li, S. Zhang, J. Hu, L. Cao, X. Hong, X. Mao, F. Huang, Y. Wu, and R. Ji, “Image-to-image translation via hierarchical style disentanglement,” in *CVPR*, 2021. 2
- [73] Y. Liu, Y. Li, S. You, and F. Lu, “Unsupervised learning for intrinsic image decomposition from a single image,” in *CVPR*, 2020. 2
- [74] T. Bepler, E. Zhong, K. Kelley, E. Brignole, and B. Berger, “Explicitly disentangling image content from translation and rotation with spatial-vae,” in *NeurIPS*, 2019. 2
- [75] X. Yang, Z. Xu, and J. Luo, “Towards perceptual image dehazing by physics-based disentanglement and adversarial training,” in *AAAI*, 2018. 2
- [76] R. Gong, D. Dai, Y. Chen, W. Li, and L. Van Gool, “Analogical image translation for fog generation,” in *AAAI*, 2021. 2
- [77] K. Garg and S. K. Nayar, “Photorealistic rendering of rain streaks,” *ACM TOG*, 2006. 2, 6, 7
- [78] Y. Weber, V. Jolivet, G. Gilet, and D. Ghazanfarpour, “A multiscale model for rain rendering in real-time,” *Computers & Graphics*, 2015. 2
- [79] P. Rousseau, V. Jolivet, and D. Ghazanfarpour, “Realistic real-time rain rendering,” *Computers & Graphics*, 2006. 2
- [80] P. C. Barnum, S. Narasimhan, and T. Kanade, “Analysis of rain and snow in frequency space,” *IICCV*, 2010. 2
- [81] C. Sakaridis, D. Dai, and L. Van Gool, “Semantic foggy scene understanding with synthetic data,” *IICCV*, 2018. 2, 4
- [82] M. Roser, J. Kurz, and A. Geiger, “Realistic modeling of water droplets for monocular adherent raindrop recognition using bezier curves,” in *ACCV*, 2010. 2
- [83] Z. Hao, S. You, Y. Li, K. Li, and F. Lu, “Learning from synthetic photorealistic raindrop for single image raindrop removal,” in *ICCV Workshops*, 2019. 2
- [84] S. You, R. T. Tan, R. Kawakami, Y. Mukaigawa, and K. Ikeuchi, “Adherent raindrop modeling, detectionand removal in video,” *T-PAMI*, 2015. 2
- [85] H. Porav, T. Bruls, and P. Newman, “I can see clearly now: Image restoration via de-raining,” in *ICRA*, 2019. 2, 5, 10, 11, 12
- [86] J. Gu, R. Ramamoorthi, P. Belhumeur, and S. Nayar, “Removing image artifacts due to dirty camera lenses and thin occluders,” in *SIGGRAPH*, 2009. 2
- [87] V. Muşat, I. Fursa, P. Newman, F. Cuzzolin, and A. Bradley, “Multi-weather city: Adverse weather stacking for autonomous driving,” in *ICCV Workshops*, 2021. 2
- [88] A. Lengyel, S. Garg, M. Milford, and J. C. van Gemert, “Zero-shot day-night domain adaptation with a physics prior,” in *ICCV*, 2021. 2
- [89] N. Hansen, S. D. Müller, and P. Koumoutsakos, “Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es),” *Evolutionary computation*, 2003. 4, 5
- [90] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, “Least squares generative adversarial networks,” in *ICCV*, 2017. 3, 6
- [91] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *ICCV*, 2017. 5
- [92] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liang, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “nuscenes: A multimodal dataset for autonomous driving,” in *CVPR*, 2020. 5, 6, 8, 11
- [93] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *CVPR*, 2016. 5, 6, 9
- [94] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O’Dea, M. Uricár, S. Milz, M. Simon, K. Amende et al., “Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving,” in *ICCV*, 2019. 5, 6, 10
- [95] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, “The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes,” in *CVPR*, 2016. 5, 6, 10
- [96] M. Uricar, G. Sistu, H. Rashed, A. Vobecsky, P. Krizek, F. Burger, and S. Yogamani, “Let’s get dirty: Gan based data augmentation for soiling and adverse weather classification in autonomous driving,” in *WACV*, 2021. 6, 7, 10
- [97] “Rain drops on screen.” <https://www.shadertoy.com/view/ldSBWW>. 7
- [98] J. C. Halimeh and M. Roser, “Raindrop detection on car windshields using geometric-photometric environment construction and intensity-based correlation,” in *IV*, 2009. 7
- [99] A. Cord and D. Aubert, “Towards rain detection through use of in-vehicle multipurpose cameras,” in *IV*, 2011. 7
- [100] A. P. Pentland, “A new sense for depth of field,” *T-PAMI*, 1987. 7
- [101] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” in *NeurIPS*, 2016. 8
- [102] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *CVPR*, 2018. 8, 9
- [103] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *CVPR*, 2017. 9
- [104] S. G. Narasimhan and S. K. Nayar, “Vision and the atmosphere,” *IJCV*, 2002. 9, 13



Fabio Pizzati completed his PhD at Inria Paris in 2022, supervised by Raoul de Charette. Before that, he received his MSc in Computer Engineering in 2018 and his BSc in Computer, Communication and Electronic Engineering in 2015, both from University of Parma, Italy. His research focuses on generative networks, vision in adverse weather and domain adaptation.



Pietro Cerri received the Dr. Eng. degree in computer engineering from the Università di Pavia, Pavia, Italy, in 2003, and the Ph.D. degree in information technologies from the Università di Parma, Parma, Italy, in 2007. He is currently team leader at Vislab, an Ambarella Company. His research is mainly focused on computer vision and sensors fusion approaches for the development of advanced driver assistance systems.



Raoul de Charette is a researcher at Inria Paris since 2015. He received the MSc degree in arts and computer graphics from Paris 8 Uni. and the PhD degree in computer vision for autonomous driving from Mines ParisTech in 2012. He was with Mines ParisTech (2008-2011, 2012-2013), Carnegie Mellon University (2011), and University of Makedonia (2014). He now leads the Astra-vision group in the Astra team, Inria.