Bounding Box-Guided Diffusion for Synthesizing Industrial Images and Segmentation Maps

Alessandro Simoni*, Francesco Pelosin*
Covision Lab
Brixen, South Tyrol, Italy
name.surname@covisionlab.com

Abstract

Synthetic dataset generation in Computer Vision, particularly for industrial applications, is still underexplored. Industrial defect segmentation, for instance, requires highly accurate labels, yet acquiring such data is costly and time-consuming. To address this challenge, we propose a novel diffusion-based pipeline for generating high-fidelity industrial datasets with minimal supervision. Our approach conditions the diffusion model on enriched bounding box representations to produce precise segmentation masks, ensuring realistic and accurately localized defect synthesis. Compared to existing layout-conditioned generative methods, our approach improves defect consistency and spatial accuracy. We introduce two quantitative metrics to evaluate the effectiveness of our method and assess its impact on a downstream segmentation task trained on real and synthetic data. Our results demonstrate that diffusion-based synthesis can bridge the gap between artificial and real-world industrial data, fostering more reliable and cost-efficient segmentation models. The code is publicly available at https://github.com/covisionlab/diffusion_labeling.

1. Introduction

Dataset synthesis has gained significant importance in recent years, particularly within the Natural Language Processing (NLP) community, where we witnessed major improvements in both academic and industrial applications [3, 5, 37]. These methods have proven especially valuable in scenarios where collecting and annotating real-world data is expensive or impractical.

In contrast, dataset synthesis in Computer Vision remains an emerging field and its usage is still under study [8, 9]. Its potential to reduce labeling costs and mitigate data scarcity constitute an appealing property for the deep learning paradigm. Despite its potential, the field remains relatively underexplored compared to its NLP counterpart. This is particularly true in domains where acquiring precise labeled data is both costly and time-consuming, such as industrial inspection, medical imaging, and remote sensing. In these domains, even small inaccuracies in annotation can significantly impact model performance, making synthetic data generation a compelling alternative.

Most of the recent research in synthetic data for vision has focused on text-to-image generation [20, 33, 39], leveraging generative models to create realistic visuals from textual descriptions. While these advancements have paved the way for creative applications and content generation, their direct applicability to real-world industrial settings remains limited. Industrial datasets, in particular, suffer from challenges such as class imbalances, labeling inconsistencies and high quality standards. These issues necessitate the development of tailored synthesis techniques capable of generating high-fidelity data hopefully with minimal manual intervention.

A critical challenge is the automatic creation of industrial dataset samples, where balancing efficiency with accuracy is difficult. Fully automated synthesis risks generating unrealistic or irrelevant samples, reducing the utility of the data. On the other hand, manual supervision, while improving accuracy, is often infeasible due to time and cost constraints — especially when dealing with complex imaging systems that go beyond human perception such as infrared imaging [36]. Industrial defect segmentation exemplifies this challenge, as it demands highly precise annotations to train reliable models.

To address these limitations, we propose a novel pipeline for generating realistic synthetic samples with cheap supervision. Our approach leverages diffusion models conditioned on human-provided bounding boxes to produce precise segmentation masks. By doing so, we unlock the generation of high-quality industrial datasets while exploiting human domain expertise but with a significant reduction in the burden of manual annotation.

*Equal contribution.

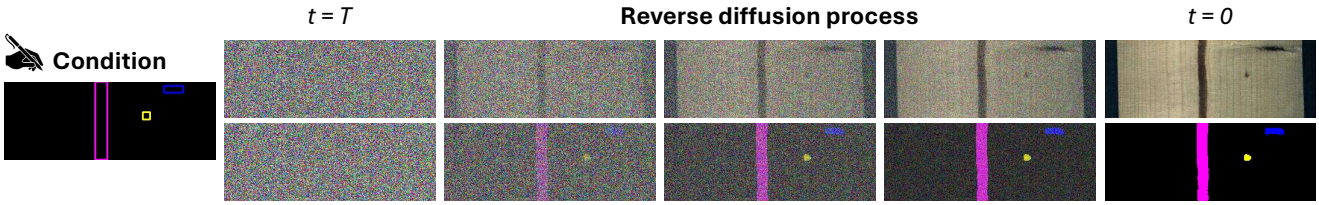

Figure 1. Overview of the proposed diffusion-based approach that generates both RGB and segmentation map in industrial setting.

In industrial settings, diffusion models have been employed for data synthesis only in classification tasks [28]. However, to the best of our knowledge and according to a recent review [41], no existing work has addressed the challenge of synthetic dataset generation for semantic segmentation, generating high quality labels from inexpensive annotation.

We present a diffusion-based approach, depicted in Figure 1, that generates RGB images and semantic maps leveraging an enriched bounding box representation as conditioning. We compare it with a modified state-of-the-art approach on layout-conditioned generation [40]. Our baseline exhibits superior consistency in generating defects within the provided bounding box annotations, making it preferable over existing generative pipelines. In this regard, we propose two metrics to quantitatively evaluate the obtained results. Ultimately, we provide some experiments showing the quality of the generated data by monitoring the performance of a downstream segmentation task trained on both real and synthetic data. Thus, we shed light on the potential of diffusion-based synthesis in bridging the gap between artificial and real-world industrial data, fostering more accurate and efficient computer vision models for segmentation.

To sum up, our main contributions are as follows:

- We introduce a novel synthetic data generation pipeline that leverages diffusion models conditioned on human-provided bounding boxes to generate high-fidelity industrial dataset samples.
- The proposed approach, thanks to an enriched bounding box representation, ensures that the generated defects remain both realistic and accurately localized within the bounding box boundary, enhancing segmentation consistency.
- By reducing the reliance on manual labeling, our method significantly lowers the cost and time required for curating industrial datasets while maintaining high annotation quality.
- We propose two metrics and evaluate our approach against a state-of-the-art conditioned diffusion pipeline, demonstrating competitive performance and improved control over defect placement.
- Our findings highlight the potential of diffusion-based dataset synthesis to improve industrial defect segmentation models, unlocking the development of more robust computer vision solutions in real-world settings.

2. Related Works

Synthetic data generation has been explored through various methodologies, each catering to specific domains and applications.

3D Game Engines. One prevalent approach leverages 3D game engines such as Unreal Engine [1], where meticulously crafted scenes or objects serve as high-fidelity proxies of reality. This method has been widely adopted, leading to the creation of extensive datasets and comprehensive frameworks [7, 25, 27], which have subsequently facilitated advancements in novel methodologies [32, 42].

GAN / Diffusion. Another powerful paradigm involves neural generative models. Techniques such as GANs [10] and diffusion models [15] have demonstrated remarkable efficacy in producing high-fidelity synthetic data. These models have found widespread applications, ranging from medical imaging [6, 11, 31], self-driving car research [22, 24], privacy preservation [17] and finally in robotics, where has been investigated for pose estimation, as discussed in [29].

Foundation Models. Recently, foundation models have also been explored for synthetic data generation. Notably, COSMOs [23] facilitates the creation of entire synthetic video sequences, while large vision-text models have been widely utilized for generative applications [20, 33, 39].

Conditioned Generation. Our pipeline not only generates RGB images but also their correspondent labels. A related study [35], proposes a method for end-to-end RGB and label generation for satellite data. While their approach is purely generative, ours allows human intervention, granting users the flexibility to place annotations as needed. This distinction enhances the control and accuracy of label generation.

Additionally, we consider [40], a generative approach that conditions data synthesis on bounding boxes. We will compare our method with this approach in later sections to provide a comprehensive evaluation of our proposed framework.

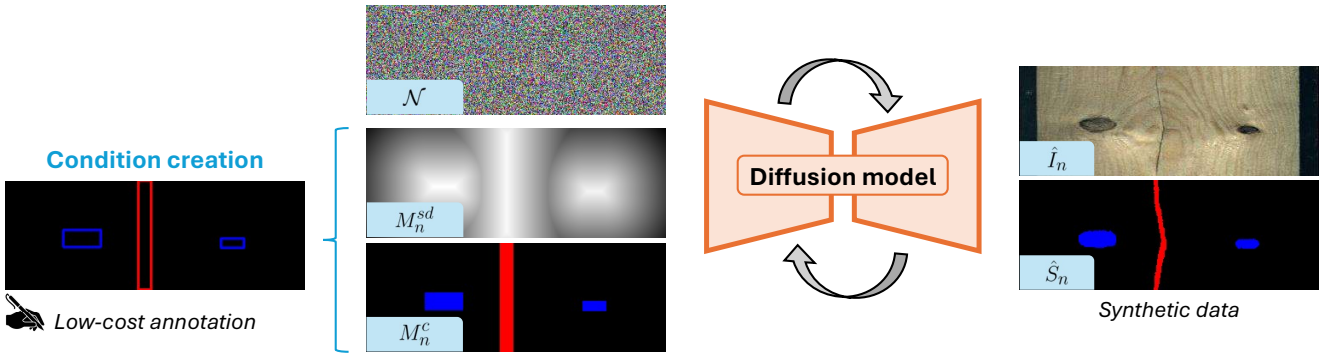

Figure 2. An overview of the proposed method: the user produces low-cost bounding box annotations which are then converted in two representations (BASD and C-BASD). Later, these encodings, are fed into the diffusion to condition the generation of both high quality RGB and segmentation masks of wood defects.

3. Method

In this section, we describe the proposed method shown in Figure 2.

3.1. Problem statement

In this work, we address the challenging task of semantic segmentation in an industrial setting. Since the lack of annotated data is very common, a way to tackle this problem is to augment the annotations with synthetic samples. Thus, we aim to adapt a conditional diffusion-based pipeline to denoise both an RGB image and its segmentation map as annotation.

Formally, we define a dataset $\mathcal{D} = \{(I_n, S_n, B_n) \mid n = 1, \dots, N\}$ where:

- $I_n^{H \times W \times 3}$ is an RGB image,
- $S_n^{H \times W}$ is the corresponding segmentation map composed of the discrete pixels values $c_{ij} \in \{1, 2, 3, \dots, C\}$ where C is the total number of classes,
- $B_n = \{b_k : (c, i_{min}, j_{min}, i_{max}, j_{max}), k = 1, \dots, K\}$ is a tuple that identifies the class of the object and its bounding box location as the top left (i_{min}, j_{min}) and bottom right (i_{max}, j_{max}) corners.

Our method applies the diffusion process to the couple (I_n, S_n) conditioned on B_n. In the following, we thoroughly describe how we preprocess the inputs and the training pipeline of the proposed method.

3.2. Data preprocessing

The first step is to process the segmentation map S_n and the bounding boxes B_n to allow the diffusion process to work with continuous values.

Segmentation map. Since the goal is to generate synthetic samples according to the joint probability $p(I_n, S_n)$, we need to make sure that these data are in the same continuous space $\mathbb{R}$. Drawing inspiration from [4, 35], we convert the segmentation map into an analog bit representation. Formally, the pixelwise discrete segmentation values c_{ij} are mapped into a binary code defined as

$$bin : \{1, 2, 3 \dots, C\} \rightarrow \{0, 1\}^{\lceil \log_2 C \rceil} \tag{1}$$

After this encoding, the segmentation map dimension is $H \times W \times \lceil \log_2 C \rceil$. As proven by previous works [4], this representation is more effective than one-hot encoding which is also less efficient in terms of number of channels in the presence of a high number of classes C. After the binary encoding, a normalization is applied to change the range from $[0, 1]$ to $[-1, 1]$ which is the same of the RGB image I_n.

Bounding box. To condition the generation of the synthetic couple $(\hat{I}_n, \hat{S}_n)$ on the bounding boxes, we create an enriched representation of B_n that encodes both spatial and class information. The spatial information is captured in terms of pixelwise encoding. Thus, we compute a Bounding Box-Aware Signed Distance (BASD) map M_n^d that assigns to each pixel (i, j) the minimum distance to the nearest bounding box boundary point. The distance value is positive inside a bounding box and negative outside. Moreover, a Bounding-Box Class (C-BASD) map M_n^c is computed accordingly assigning to each positive value the corresponding class of the boundary point. We formally define the computation of M_n^d and M_n^c in Algorithm 1 and a visualization of the resulting maps can be seen in Figure 2.

Before concatenating these two representation maps to the couple (I_n, S_n), the class map M_n^c is encoded with the previously introduced analog bit paradigm obtaining an output dimension of $H \times W \times \lceil \log_2 C \rceil$. Our encoding assigns a single class per pixel but still handles overlapping bounding boxes. When two boxes overlap, the class map forms a structured pattern reflecting the overlap location instead of arbitrarily selecting one class. This allows the network to learn spatial relationships without needing explicit multi-label assignments, which a pure analog bit encoding can not achieve.

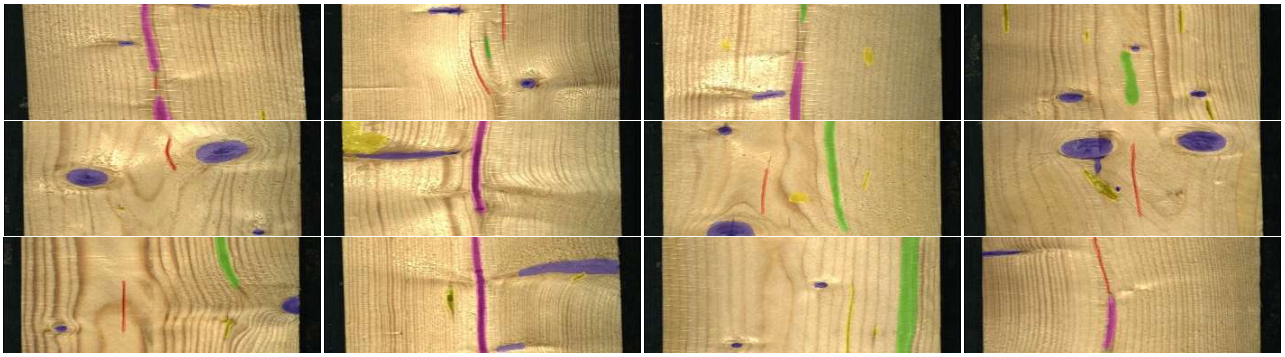

Figure 3. Some samples of the Wood Defect Detection [18] with the semantic segmentation labels. The wood defects are the following: knot (blue), crack (red), quartzite (green), resin (yellow), marrow (magenta).

Algorithm 1 M_n^d and M_n^c computation. Comments in blue.

Require: Bounding boxes B_n
Ensure: M_n^d of size (H, W), M_n^c of size (H, W)
Initialize $M_n^d \leftarrow +\infty$ for all pixels p_{ij}
Initialize $M_n^c \leftarrow 0$ for all pixels p_{ij}
for each $b_k \in B_n$ with class c **do**
  Compute boundary pixels of b_k:
  $\beta \leftarrow \text{Boundary}(b_k)$
  for each pixel p_{ij} **do**
    Compute distance to the closest boundary point:
    $d_\beta \leftarrow \min_{(i_\beta, j_\beta) \in \beta} \sqrt{(i - i_\beta)^2 + (j - j_\beta)^2}$
    $d_\beta \leftarrow d_\beta * \text{InOutSign}(p_{ij}, b_k)$
    Update M_n^d and M_n^c:
    if $|d_{\beta_n}| < |M_n^d(p_{ij})|$ **then**
      $M_n^d(p_{ij}) \leftarrow d_\beta$
      $M_n^c(p_{ij}) \leftarrow c$

3.3. Conditioned Diffusion Model

To synthesize realistic and structurally consistent images, we condition the denoising diffusion process on our enriched bounding box representation. A UNet architecture takes as input $(x_0, (M_n^d, M_n^c))$ where $x_0 = (I_n, S_n)$. The output is the couple $(\hat{I}_n, \hat{S}_n)$ comprising of an RGB image plus its segmentation map with dimension $H \times W \times 3 + \lceil \log_2 C \rceil$.

Given a clean sample x_0, the forward diffusion process gradually adds Gaussian noise:

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\alpha_t}\, x_0,\, (1 - \alpha_t) I), \tag{2}$$

where α_t is the noise scheduling coefficient. The reverse process learns to reconstruct x_0 while incorporating the structural constraints from the conditioning (M_n^d, M_n^c):

$$\begin{aligned} &p_\theta(x_{t-1} \mid x_t, M_n^d, M_n^c) = \\ &\mathcal{N}\big(x_{t-1}; \mu_\theta(x_t, t, M_n^d, M_n^c), \sigma_t^2 I\big). \end{aligned} \tag{3}$$

where $\mu_\theta(x_t, t, M_n^d, M_n^c)$ is the predicted denoised estimate and σ_t is the variance of the noise distribution.

The diffusion model is trained by minimizing the noise prediction loss:

$$\mathbb{E}_{x_0, M_n^d, M_n^c, t, \epsilon} \left[\|\epsilon - \epsilon_\theta(x_t, t, M_n^d, M_n^c)\|^2 \right], \tag{4}$$

with $\epsilon \sim \mathcal{N}(0, I)$ representing the injected Gaussian noise. This formulation ensures that the generated samples adhere to both the semantic structure encoded in the segmentation and the spatial constraints provided as bounding box conditioning.

4. Experiments

In this section, we discuss the implementation details and the industrial dataset we used for our experiments. Finally, a thorough comparison between our approach and a state-of-the-art conditional diffusion model [40] is assessed in terms of quality and consistency.

4.1. Experimental setting

Diffusion model. The proposed method follows the DDPM [14] paradigm with a UNet [26] architecture trained from scratch. We modified the input and output channels accordingly to support our bounding box encoding representation and the denoising of the segmentation map. Both during training and testing the number of denoising iterations were set to 1000. We trained for 300 epochs using AdamW [21] as optimizer with learning rate $1e^{-5}$ and batch size 8 on two Nvidia RTX 4090. The total training time is approximately 1 day.

Downstream task. For the semantic segmentation downstream task we emploied a UNet architecture with a ResNet-18 [12] backbone. We used a single network for each segmentation class to avoid class balancing problems and concentrate on the synthetic data assessment. The training lasted 100 epochs using AdamW as optimizer with learning rate $1e^{-5}$ and batch size 64 on a single Nvidia RTX 4090.

Data	FID ↓ @2048	@768	@192	@64	KID ↓ @2048	@768	@192	@64	LPIPS ↓ *AlexNet*	*VGG-16*	*SqueezeNet*
Synth [40]	**40.94**	**0.25**	24.04	6.77	**40.94**	**8.07**	19×10^3	10×10^3	0.35	0.49	0.26
Synth *Ours*	45.47	0.30	**14.49**	**3.09**	45.46	8.73	$\mathbf{10 \times 10^3}$	$\mathbf{3.6 \times 10^3}$	**0.28**	**0.43**	**0.21**

Table 1. Assessment of generation quality. We report the FID and KID computed at different levels of the InceptionV3 [34] network, and the LPIPS computed with several backbones AlexNet [19], SqueezeNet [16] and VGG-16 [30].

Method	SAE (%) ↓ Knot	Crack	Quartzite	Resin	Marrow	**Avg**
Layout Diffusion [40]	40.03	83.82	61.00	85.59	54.88	46.77
Ours	**5.53**	**4.57**	**3.19**	**4.82**	**3.64**	**4.99**

Table 2. Comparison between our method and [40] in terms of Segmentation Alignment Error. The Avg is computed over all pixels.

Method	EBR (%) ↓ Knot	Crack	Quartzite	Resin	Marrow	**Avg**
Layout Diffusion [40]	13.43	69.16	48.76	80.15	28.44	26.00
Ours	**0.86**	**2.41**	**4.98**	**2.22**	**0.89**	**5.51**

Table 3. Comparison between our method and [40] in terms of Empty Bounding-Box Rate. The Avg is computed over all bounding boxes.

Dataset. Since we focus on the industrial setting, we selected the Wood Defect Detection [18] dataset, a semantic segmentation and object detection collection of data for the wood manufacturing industry. It contains 20276 images with semantic segmentation and bounding box annotations of 10 different classes of wood defects. In our experiments, we decided to aggregate the 4 classes of knots and avoiding the classes of blue stain and overgrown that are underrepresented. Thus, we obtained a dataset comprising of 20107 images with a total of 5 defect classes (knot, crack, quartzite, resin, marrow).

Moreover, we split the dataset into three subsets: 70% for training the diffusion model, 20% for training the segmentation model, and 10% as a fixed real test set. Additionally, the bounding box annotations from the 20% real split are used to generate synthetic data for evaluating the semantic segmentation task. Figure 3 illustrates some samples from the original dataset.

4.2. Data synthesis assessment

To assess the quality of synthetic data, we compare our approach with the current state-of-the-art layout-conditional diffusion model [40], utilizing its original code implementation and adapting it to take non-squared images. Specifically, we focus on evaluating the consistency between the generated defects and their corresponding bounding box constraints. To quantify this relationship, we introduce two metrics, the Segmentation Alignment Error (SAE) and the Empty Bounding-Box Rate (EBR).

Segmentation Alignment Error (SAE). With this measure we quantify how many generated defect pixels fall outside their designated bounding boxes, indicating misalignment between the generated defects and their constraints. Formally, let:

- $\hat{P}$ be all the generated pixels of segmented defects,
- $\hat{P}_{out}$ be the generated pixels that fall outside the bounding boxes.

Thus, we define the metric as follows:

$$\text{SAE} = \frac{\hat{P}_{out}}{\hat{P}} \tag{5}$$

where a lower value indicates that the model is more consistent with the generation condition.

As shown in Table 2, the method proposed in [40] struggles to maintain defect placement within the bounding boxes, resulting in a very high mean SAE of 46.77% across all the defects. In contrast, our approach, leveraging a dual bounding box encoding strategy (BASD and C-BASD), significantly improves alignment, with only 4.99% of generated pixels falling outside the given regions.

Empty Bounding-Box Rate (EBR). To assess whether the generated defects correctly fall within their designated bounding boxes, we define the Empty Bounding-Box Rate (EBR). This metric quantifies how many bounding boxes remain empty, meaning no synthetic pixels are generated inside them. Formally, let:

- $B_{all} = \{b_k \mid b_k \in B_n \ , \ n = 1, \dots, N\}$ be the set of all bounding boxes,
- $B_{miss} = \{b_k \mid b_k \in B_{all} \ , \ \mathcal{G} \cap b_k = \emptyset\}$ be the subset of bounding boxes that contain no generated pixels.

Thus, we define the metric as follows:

$$\text{EBR} = \frac{|B_{miss}|}{|B_{all}|} \tag{6}$$

where higher values indicate that a larger number of bounding boxes have been missed during generation, signifying a poorer retrieval of the provided conditioning.

As reported in Table 3, the EBR metric shows the superiority of our proposal in retrieval abilities by a large margin. Specifically, our average EBR lies around 5.51% on the total amount of bounding boxes and surpasses by more than 20% points the competitor [40].

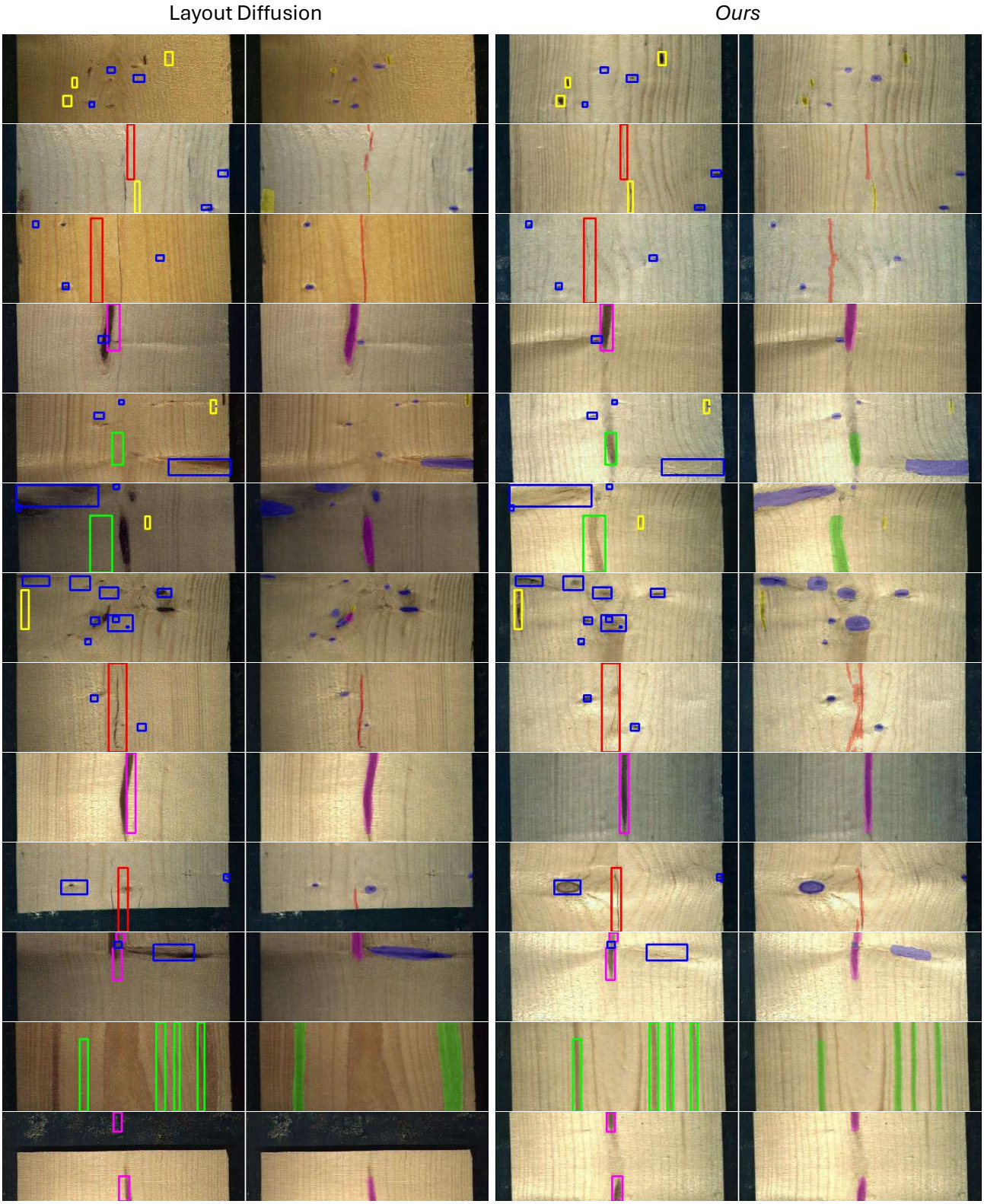

Figure 4. Qualitative comparison between our method and Layout Diffusion [40]. Each method shows the generated RGB image with respect to the bounding box condition - the same for both methods - and the overlapped defect segmentation map. The wood defects are the following: knot (blue), crack (red), quartzite (green), resin (yellow), marrow (magenta).

Train data	F1 (%) ↑ Knot	Crack	Quartzite	Resin	Marrow	Avg
Real	78.56	48.80	24.49	45.00	65.40	52.45
Synth [40]	72.15	8.20	21.03	18.01	58.04	35.49
Synth *Ours*	76.57	45.56	12.82	32.71	58.18	45.17
Real+Synth [40]	78.44	46.71	**27.01**	43.85	**71.09**	53.42
Real+Synth *Ours*	**79.48**	**50.38**	25.85	**46.29**	66.11	**53.62**

Table 4. Downstream task assessment in terms of F1 score using real, synthetic and real+synthetic data during training.

Visual sample quality. To further analyze the quality of the generated synthetic images, we report the Fréchet Inception Distance (FID) [13], the Kernel Inception Distance (KID) [2] and the LPIPS [38] metrics. As can be observed in Table 1, our method tends to have better performance on low-level features with regard to the FID and KID metrics, meaning that the local perception of the details is better than the competitor. Additionally, our method outperforms [40] across all tested backbones in terms of LPIPS metric, confirming that the generated images exhibit higher perceptual realism across different network architectures.

Qualitative results. To further illustrate this comparison, Figure 4 depicts qualitative examples. Moreover, the results demonstrate that [40] not only fails to confine defects within the bounding boxes but also occasionally generates wrong segmentation labels.

4.3. Downstream task evaluation

To evaluate the effectiveness of our synthetic data, we conduct a semantic segmentation experiment using a UNet architecture trained on different data configurations.

Starting from the 20% split, we use the original bounding box annotations as guidance to generate couples of image and label. We do so for both methods, ours and [40]. We then use this synthetic split to train the segmentation pipeline. Moreover, to ensure a fair comparison between approaches, we discard synthetic pixel labels generated outside the bounding boxes conditioning. This prevents eventual generalization of the downstream segmentation given by extra synthetic labels generated without explicit conditioning.

Table 4 presents the F1 scores computed on the 10% real test split, where we compare models trained on real data, synthetic data, and a combination of both. Notably, when training on synthetic data alone, our approach surpasses [40] by an impressive 10%, demonstrating its ability to generate more valid training samples. This highlights the superior quality and consistency of our synthetic segmentation maps, which provide a more reliable learning signal for the segmentation task.

When incorporating real data into the training process, the performance gap between the two methods narrows, as real samples provide a strong baseline. However, even in this hybrid setting, leveraging our synthetic data leads to the best overall F1 score, achieving a $+1.17\%$ improvement over using only real data. This result underscores the effectiveness of our method in complementing real-world annotations, reinforcing its practical utility in industrial applications where obtaining high-quality segmentation labels can be costly and time-consuming.

5. Conclusion

In this work, we are the first [41] to study the problem of data synthesis for semantic segmentation in industrial settings, where quality and precision is of importance. We devised a pipeline to generate synthetic RGB data and its segmentation label counterpart at the same time, starting from bounding box conditioning. This allows to decrease significantly the labeling costs while preserving the quality of the segmentation maps.

We validated the performances of our method by comparing our proposal with the current state-of-the art methodology adapted for the setting. We also assessed the quality of our generation through a downstream task, training a UNet with a combination of real and synthetic data.

The experiments suggest that our proposal is robust to spatial consistency generation, improving the performance of the downstream segmentation task.

We also introduced dedicated metrics useful for the community to assess the correctness of layout-conditioned data generation.

References

[1] Unreal Engine. https://www.unrealengine.com. 2

[2] Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations (ICLR)*, 2018. 7

[3] Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*, 2024. 1

[4] Ting Chen, Ruixiang Zhang, and Geoffrey E. Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *International Conference on Learning Representations (ICLR)*, 2023. 3

[5] Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In *International Conference on Machine Learning (ICML)*, 2024. 1

[6] Virginia Fernandez, Walter Hugo Lopez Pinaya, Pedro Borges, Petru-Daniel Tudosiu, Mark S. Graham, Tom Vercauteren, and M. Jorge Cardoso. Can segmentation models be trained with fully synthetically generated data? In *Simulation and Synthesis in Medical Imaging (MICCAI Workshops)*, 2022. 2

[7] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking anal-

ysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[8] Scott Geng, Ranjay Krishna, and Pang Wei Koh. Training with real instead of synthetic generated images still performs better. In *Synthetic Data for Computer Vision Workshop (CVPRW)*, 2024. 1

[9] Magnus Kaufmann Gjerde, Filip Slezák, Joakim Bruslund Haurum, and Thomas B Moeslund. From nerf to 3dgs: A leap in stereo dataset quality? In *Synthetic Data for Computer Vision Workshop (CVPRW)*, 2024. 1

[10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial networks. *Communications ACM*, 2020. 2

[11] Pengfei Guo, Can Zhao, Dong Yang, Ziyue Xu, Vishwesh Nath, Yucheng Tang, Benjamin Simon, Mason Belue, Stephanie Harmon, Baris Turkbey, et al. Maisi: Medical ai for synthetic imaging. *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025. 2

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems (NIPS)*, 2017. 7

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NIPS)*, 33, 2020. 4

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NIPS)*, 2020. 2

[16] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 5

[17] Marvin Klemp, Kevin Rösch, Royden Wagner, Jannik Quehl, and Martin Lauer. LDFA: latent diffusion face anonymization for self-driving applications. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[18] P Kodytek, A Bodzas, and P Bilik. A large-scale image dataset of wood surface defects for automated vision-based quality control processes. *F1000Research*, 2022. 4, 5

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012. 5

[20] Baiqi Li, Zhiqiu Lin, Deepak Pathak, Jiayao Emily Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Genai-bench: A holistic benchmark for compositional text-to-visual generation. In *Synthetic Data for Computer Vision Workshop (CVPRW)*, 2024. 1, 2

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. 4

[22] Quang Nguyen, Truong Vu, Anh Tran, and Khoi Nguyen. Dataset diffusion: Diffusion-based synthetic data generation for pixel-level semantic segmentation. In *Advances in Neural Information Processing Systems (NIPS)*, 2023. 2

[23] NVIDIA. Cosmos world foundation model platform for physical ai, 2025. 2

[24] Ethan Pronovost, Meghana Reddy Ganesina, Noureldin Hendy, Zeyu Wang, Andres Morales, Kai Wang, and Nick Roy. Scenario diffusion: Controllable driving scenario generation with diffusion. In *Advances in Neural Information Processing Systems (NIPS)*, 2023. 2

[25] Alexander Raistrick, Lahav Lipson, Zeyu Ma, Lingjie Mei, Mingzhe Wang, Yiming Zuo, Karhan Kayan, Hongyu Wen, Beining Han, Yihan Wang, Alejandro Newell, Hei Law, Ankit Goyal, Kaiyu Yang, and Jia Deng. Infinite photorealistic worlds using procedural generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*. Springer, 2015. 4

[27] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[28] Thanyathep Sasiaowapak, Siridech Boonsang, Santhad Chuwongin, Teerawat Tongloy, and Pattarachai Lalitrojwong. Generative ai for industrial applications: Synthetic dataset. In *International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2023. 2

[29] Alessandro Simoni, Stefano Pini, Guido Borghi, and Roberto Vezzani. Semi-perspective decoupled heatmaps for 3d robot pose estimation from depth maps. *IEEE Robotics and Automation Letters (RAL)*, 2022. 2

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 5

[31] Youssef Skandarani, Pierre-Marc Jodoin, and Alain Lalande. Gans for medical image synthesis: An empirical study. *J. Imaging*, 2023. 2

[32] Chunyi Sun, Junlin Han, Weijian Deng, Xinlong Wang, Zishan Qin, and Stephen Gould. 3d-gpt: Procedural 3d modeling with large language models. *arXiv preprint arXiv:2310.12945*, 2023. 2

[33] Jiao Sun, Deqing Fu, Yushi Hu, Su Wang, Royi Rassin, Da-Cheng Juan, Dana Alon, Charles Herrmann, Sjoerd van Steenkiste, Ranjay Krishna, et al. Dreamsync: Aligning text-to-image generation with image understanding feedback. *arXiv preprint arXiv:2311.17946*, 2023. 1, 2

[34] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[35] Aysim Toker, Marvin Eisenberger, Daniel Cremers, and Laura Leal-Taixé. Satsynth: Augmenting image-mask pairs through diffusion models for aerial semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3

[36] Doan Thinh Vo, Phan Anh Duc, Nguyen Nhu Thao, and Huong Ninh. An approach to synthesize thermal infrared ship images. In *Synthetic Data for Computer Vision Workshop (CVPRW)*, 2024. 1

[37] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023. 1

[38] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 7

[39] Yasi Zhang, Peiyu Yu, and Ying Nian Wu. Object-conditioned energy-based model for attention map alignment in text-to-image diffusion models. In *Synthetic Data for Computer Vision Workshop (CVPRW)*, 2024. 1, 2

[40] Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 4, 5, 6, 7

[41] Hans Aoyang Zhou, Dominik Wolfschläger, Constantinos Florides, Jonas Werheid, Hannes Behnen, Jan-Henrick Woltersmann, Tiago C. Pinto, Marco Kemmerling, Anas Abdelrazeq, and Robert H. Schmitt. Generative AI in industrial machine vision - A review. *arXiv preprint arXiv:2408.10775*, 2024. 2, 7

[42] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2023. 2