EC4N36

Methodological Guideline for NeuralDE

Document reference number for ANR

contact@confiance-ai.fr | www.confiance.ai

Document reference: XXX

Contributors

	Name	Organisation	Role
Responsible for the deliverable	Martin Gonzalez	IRT SystemX	AS Lead
Scientific responsible	Martin Gonzalez	IRT SystemX	AS Lead
Co-authors	Nelson Fernandez Pinto	Air Liquide	FA Contributor

Document control

Revision	Date	Commentary	Author
1.0	XX	XX	XX
00	XX	XX	XX
00	XX	XX	XX
00	XX	XX	XX
00	XX	XX	XX

A. Introduction

Trustworthiness in AI within critical systems (systems that can directly or indirectly affect human life and moral entities) is essential for its widespread adoption (by the industry, the decision makers, the general public, etc.) and poses the following significant challenges:

- First, how to design AI models, so that, by construction, they satisfy trustworthy properties (accuracy, robustness…).
- Secondly, how to characterize these AI models, for example to understand and explain their behavior and their adequacy to the operational domain.
- Then, how to implement and embed those AI models on hardware, by making them fit for the target without losing their trustworthy properties.
- Another question is, what methods of data engineering to apply in order to, among other topics, manage important volumes of data and adapt to the evolution of the operational domain.
- At system level, what verification and certification processes to consider specifically for AI-based systems.
- Finally, a federation of all these matters is necessary to build an end-to-end methodological approach, supported by a consistent engineering environment compatible with industrial practices.

These are the challenges, among others, that the Confiance.ai program addresses.

Trustworthy AI Challenges

To learn, machine learning models need to train on an immense amount of data. Their learning must, in principle, cover all possible scenarios. But unpredictable scenarios inexorably appear in an industrial context where human and environmental factors intervene. In addition, some data is difficult to obtain and use. The challenge therefore lies in the ability to propose machine learning models that maintain their operation in scenarios for which no or little real data is available. Synthetic data, produced artificially by generative AI approaches, thus represents an effective response to this problem and a means, for industrial players, to create more efficient and more robust AI-based systems.

Rationale of this Methodological Guideline

The rationale behind this document is to provide the users of the NeuralDE component with a comprehensive guide on how to understand the NeuralDE component within the Confiance.ai ecosystem, and to provide good practices on how to use it at an industrial level, based on the Use-Case experiences of usage of it.

Target Audience

The target audience of this document consists of Data and Machine Learning Engineers.

Document Organization

This document is organized as follows.

- In this Introduction, we set up common Confiance.ai challenges that this guideline addresses, we present the rationale of this document, its target audience, organization and usage.
- In Chapter B, we present an overall vision of the proposed method.
- In Chapter C, we provide an overview of the scientific concepts that are introduced in the guideline.
- In Chapter D, we proceed to describe the NeuralIDE component.
- In Chapter E, we present the two demonstration notebooks that are appended to this guideline, and which serve as a hands-on user-oriented description of each proposed method.
- In Chapter F, we provide conclusions, limitations and perspectives.

How to use this document

This document may be used in the following two manners:

- As a documented support for understanding the high-level ideas and general context of the NeuralIDE component, as well as a unified view of the three user-oriented notebooks and their individual perimeter and objective.
- If you are not familiar with test-time adaptive defences for empirical robustness, you can use it as an introductory note on test-time adaptation, which is a less known ML subject, which is essential to understanding how to use and what to expect from the component and its follow-up versions.

B. Presentation of the Methodological Guidelines

In this section, we present the NeuralDE component from the view-point of the Confiance.ai listed scientific challenges & trustworthiness attributes.

Scientific Challenges

The NeuralDE component consists on a local empirical robustification method. As such, it addresses the **Robustness** scientific challenge and the **Domain Adaptation without Bias Transfer** as a sub-challenge.

Trustworthiness Attributes

The NeuralDE component acts as a test-time adaptation method for robustifying a target pretrained model. It handles different modes of robustness which address **inherent AI local** and **global robustness**, **System-Dependent Adaptability** and **Resilience**.

Overall Vision of the NeuralDE Component

The objective of NeuralDE is to decompose robustness into three different scenarios: Adversarial Robustness, Robustness against Common Corruptions and against Distribution Shifts. The specific approach is to approach this problem from the view-point of Domain Adaptation in Test-Time (meaning without any modification of the initial pretrained model). Concretely, we propose three generic approaches for each mentioned scenario:

- **Adversarial purification** as a test-time adaptive defense against adversarial examples: this consists on annihilating adversarial noise from inputs before sending them to the target model
- **Image Restoration** as a test-time adaptive defense against common corruptions: this consists on removing the corruption context:
    - Remove noise for noise-based corruptions,
    - Remove raindrops, snowflakes, fog, etc., for adverse meteorological weather corruptions
    - Restore high-quality data for digital corruptions: JPEG compression, contrast, etc.
- **Domain Shift Inversion** as a test-time adaptive defense against distribution shifts: this consists on shifting data back to the original context (for instance, translating images by night into images by day if the target model was trained on day-time images).

The importance of Test-Time adaptative defenses lies at conception time of the overall pipeline of the Ai-based system to be deployed, after the target ML model has been trained under *normal conditions* and after gathering Real-world data to test the trained model upon.

In short, the added value of NeuralDE is to propose generic test-time adaptation methods to improve robustness against different scenarios, with adapted performance metrics, and the possibility of using it as a simulator of scenarios to which test the trained model's robustness aspects.

Disclaimer

NeuralDE methods are generic, but solutions are a compound of methods that may vary from UC to another.

C. Scientific Overview

In an industrial context involving IA-based systems whose underlying Machine Learning tasks are tied to Computer Vision, there are three main and different situations in where Robustness takes different definitions and are evaluated in different ways. In this section, we provide a top-down approach for motivating the test-time adaptive defense approach, as a task lying in the family of domain adaptation ML tasks, and its relations with empirical robustification techniques. We refer the reader to the State-of-the-Art document provided in the Confiance.ai program.

C.1 An overview of Robustness challenges in Computer Vision

Adversarial robustness, robustness to common corruptions, and robustness to distribution shifts are three distinct aspects of model robustness in machine learning, each addressing different types of challenges. Here's an explanation of the differences between these concepts.

Adversarial Robustness

- **Objective:** Adversarial robustness refers to a model's ability to resist adversarial attacks. Adversarial attacks involve making small, carefully crafted perturbations to input data with the goal of causing the model to make incorrect predictions. These perturbations are often imperceptible to humans.
- **Challenges:** Adversarial attacks can exploit vulnerabilities in a model's decision boundaries, leading to misclassifications. Robustness against such attacks involves training models that can withstand these perturbations and maintain their accuracy.
- **Techniques:** Techniques like adversarial training or input purification are used to improve a model's adversarial robustness by incorporating adversarial examples during training.

Robustness to Common Corruptions

- **Objective:** Robustness to common corruptions focuses on a model's ability to perform well when the input data is corrupted or distorted in common ways. These corruptions can include noise, blurriness, occlusions, or other forms of degradation that may occur in real-world scenarios.
- **Challenges:** Real-world data is often noisy or imperfect, and models trained on clean data may struggle when faced with corrupted inputs. Robustness to common corruptions aims to ensure that models remain accurate and stable even in the presence of such disturbances.
- **Techniques:** Data augmentation techniques, such as introducing random noise or applying transformations to the training data, can help improve a model's robustness to common corruptions.

Robustness to Distribution Shifts

- **Objective:** Robustness to distribution shifts refers to a model's ability to generalize well across different data distributions. In practice, the distribution of data in the real world can change over time or vary across different application scenarios, and models need to adapt to these changes.
- **Challenges:** When a model is trained on one data distribution (e.g., data from one geographical location) and tested on another (e.g., data from a different location), it may suffer from a drop in performance due to the distribution shift. Robustness to distribution shifts involves making models more adaptive to these changes.
- **Techniques:** Domain adaptation, transfer learning, and techniques like fine-tuning on target domain data can help improve a model's robustness to distribution shifts by aligning the model's representations with the target data distribution.

In summary, adversarial robustness focuses on resisting deliberately crafted attacks on a model, robustness to common corruptions deals with handling noise and degradation in input data, and robustness to distribution shifts involves ensuring that models can generalize well to different data distributions. These aspects of robustness are all important in real-world machine learning applications and may require different techniques and strategies to address effectively.

C.2 Domain Generalization versus Domain Adaptation

As explained in the last section, the techniques employed for improving ML robustness in these three scenarios can be ranged into two categories:

- those that are training based: adversarial training or data augmentation
- those that are training-free: adversarial purification and image restoration or domain adaptation

At the heart of this distinction, there is a choice whether or not to retrain a model on specific data. Indeed, data augmentation serves to train a model so that it performs well on a specified target domain. Such domain can be out-of-distribution with respect to the initial training data but it can also be in-distribution. For instance, adversarial training specifies a set of threat models and trains by producing adversarial attacks and uses them as data augmentation at training time.

As we will see below, training-based methods will mostly contribute to *domain generalization* while training-free methods will mostly contribute to *domain adaptation.*

Domain generalization and domain adaptation are two techniques in machine learning that address the challenges posed by differences in data distributions between training and test datasets. They are both used to improve the model's ability to perform well on unseen data, but they differ in their approaches and goals.

Domain Generalization

- **Objective:** Domain generalization aims to train a model that can perform well across multiple, diverse domains or datasets, even when those domains were not seen during training. In other words, it focuses on making the model generalize better to entirely new and unseen data distributions.

- **Data Assumption:** Domain generalization assumes that the training dataset is composed of data from various domains, and the goal is to build a model that can learn common patterns or features across these domains while ignoring domain-specific variations.

Domain Adaptation

- **Objective:** Domain adaptation aims to adapt a model trained on one source domain (or dataset) to perform well on a target domain that may have a different data distribution. The goal is to make the model transfer its knowledge from the source domain to the target domain effectively.
- **Data Assumption:** Domain adaptation assumes that there are two or more domains involved: a source domain (where labelled data is available for training) and a target domain (where the model needs to perform well but has limited labelled data or none at all).

As such, domain generalization aims to retrain models that can generalize across multiple domains without any specific adaptation, while domain adaptation focuses on adjusting a pretrained model on one domain to perform well on a different, specific target domain. Both techniques help address issues related to domain shift and improve model performance in practical, real-world scenarios where data distributions may vary.

C.3 Test-Time adaptation as an empirical robustification technique

While domain generalization extends the number of scenarios in which a model can be expected to perform well, one cannot think in advance of the totality of scenarios that might still make the trained model vulnerable:

- An adversarially trained model can be brittle to new and unseen adversarial attacks, coming from different threat models than those that were used during training
- A model trained on a targeted data domain can be brittle against new and unforeseeable scenarios

If any of these vulnerabilities appear, it can be expensive, lengthy or simply impossible to retrain a model so that it generalizes to such domain. The main advantage of test-time adaptation is that it aims at handling exactly such situation without the need of interrupting the service by shutting down the faulty AI-based system.

Input Pre-processing

Input pre-processing is a test-time adaptation technique which has shown to be very performant recently. It consists on prepending an image-to-image model (a model that takes images as inputs and generates images as outputs), that is independent from the initial CV model, in order to pre-process images coming from novel scenarios in such a way that their outputs are well handled by the CV model. The whole pipeline (image-to-image model + CV model) can then be considered as improving the system's robustness in test-time, by making it adaptive to such novel scenarios. As such, input pre-processing is supposed to make the system, as a whole, resilient to unseen and adverse conditions. The initial model then depends on the pre-processing model to guarantee safe usage.

TO REMEMBER

Input Pre-processing as a Test-time adaptive defense does not improve the model's intrinsic robustness

If the input preprocessing model is taken away from the pipeline, then model is as brittle to new scenarios as it was initially.

D. Description of the method

D.1 Context of the Methodological Guideline

This methodological guideline lies in the context of the conception phase of a Machine Learning pipeline, according to the End-to-End Approach document. More specifically, it is a guideline involving the definition of a training domain of a ML model, specifies an adaptation domain for that model, and involves engineering an input preprocessing system to address the specific need of test-time adaptation.

D.2 Prerequisites for using this Guideline

As human-level pre-requisites:

- The user needs to have moderate familiarity with Machine Learning for Computer vision tasks.
- The user should have overviewed through the 2022 EC4 Deliverable on Robustness, in particular on the side of evaluation of empirical robustness and local empirical robustification techniques.
- Although it is not required, some familiarity on Image-to-Image translation tasks should be desirable, as well as the challenges it poses in terms of unavailable data, unlabeled data, supervised, unsupervised and semi-supervised image-to-image learning algorithms.

As context level pre-requisites:

- The problem needs to involve computer vision tasks (classification, semantic segmentation, object detection, etc.).
- A well specified domain to which the model needs to adapt, domain which involves content that is desired to be removed from the target adaptation domain.

D.3 Level of Maturity of this Guideline

The maturity of this guideline is 3 according to this maturity requirements document. In particular,

- It has been validated on the principal Use-Case Cylinder Counting from Air Liquide.
- We provide critical analysis on the secondary Use-Case Welding Inspection from Renault, as well as interest of the component on the latter.

D.4 Procedure

The overall picture of the described procedures can be illustrated in as follows:

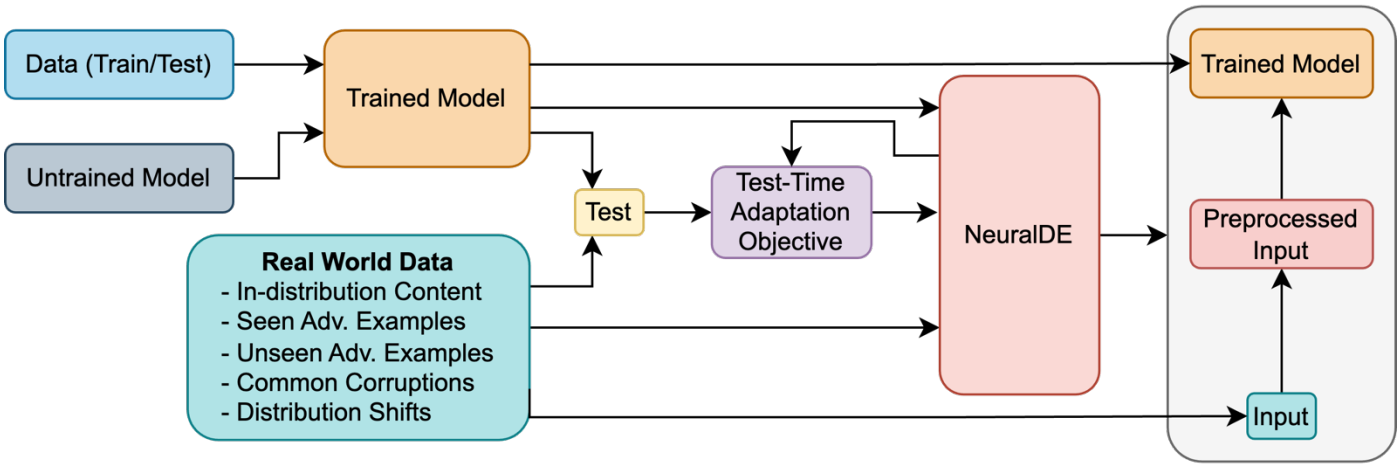

Figure 1: A diagram overview of the Procedure

The procedure involves three different steps.

1. An ML Engineer trains a model and specifies the source domain defining the training set.

   Input: training and In-Distribution test data.

   Output: trained (target) model.

2. While testing the model, the engineer identifies the nature of the domain to which the model is desired to be adaptative. In particular they identify the following modes:

   a. Adversarial: unseen adv examples (i.e., threat models unseen during training).

   b. Corruptions: presence and type of corruption.

   c. Distribution Shifts: definition of novel scenarios that are not strictly common corruptions.

      i. Additionally, the engineer must specify if the novel scenario should be included as part of training context for future iterations of model training.

   Input: Trained model and Real-world test data.

   Output: Well-defined initial test-time adaptation objective.

3. The engineer employs the NeuralDE component to create and implement a test-time adaptation pipeline for the model. The user:

   a. identifies one or a sequence of test-time adaptation models from NeuralDE;

   b. uses this pipeline as input pre-processing before feeding the pre-processed input to the target model;

   c. monitors the adaptation performance (measured by target model's performance on the real-world dataset) to calibrate the adaptation pipeline parameters optimally;

d. prepends the optimized pipeline to the target model.

Input: Trained model, test-time adaptation objective and Real-world test data.

Output: An optimised pipeline for test-time adaptation for a target model, with respect to a specific test-time adaptation objective.

Additional steps

Although the following is not directly taken in charge directly by NeuralDE, it is worth mentioning it as it completes the procedure cycle and indicates when/how to follow it in further iterations. If a novel scenario was specified to be included (c.f., 2.c.i) in the next training iteration:

- The involved actors evaluate the cost/benefit of retraining the model under an extended training set for domain generalisation versus the cost/benefits of prepending to the target model the adaptation pipeline.
- Repeat the procedure if the target model is chosen to be retrained.

E. Presentation of the demonstrators

In order to better present an operational description of each method contained in Neural DE both we deliver three comprehensive notebooks, each one serving different and complementary purposes. It is worth mentioning that part of the presented methods in these notebooks were the subject of a research article [Py, et al. 2023].

E.1 Library Demonstrator

The objective of the neuralde-examples notebooks is to present the available generic methods supported by the NeuralDE component. These are methods which are deemed to be not Use-Case specific and have been fully integrated and supported in the NeuralDE library. At the time of writing, there are 3 different methods, one for each robustness modality (adversarial, corruptions, distribution shifts), and the purpose of the notebook is to guide step-by-step the user towards demonstrating their effects on input images, highlighting a methodology to calibrate their parameters according to the test-time adaptation objective in order to best specialize it to a specific use-case. Since the purpose of this notebook is to show the functionalities of the library, we freely choose images both from public datasets as well as Confiance.ai Use-Cases, and show one by one the methods without giving details as of how to compose/concatenate them.

FOCUS

On evaluating with occlusions

It is worth noticing that the available methods correspond to scenarios in which occlusion does not play a relevant role at the preprocessing level. Nonetheless there may be scenarios in which corruptions such as snow might occlude critical zones of the image such as entire objects to be detected. In such case, evaluating the pertinence of the NeuralDE component must include additional metrics that assess the degree to which the preprocessing synthetically generated imaginary objects or deleted occluded objects.

E.2 Primary Use-Case level Demonstrator

Understanding a scene using a camera

Air Liquide uses cameras equipped with an AI-based system for automatic outdoor inventory counting. Used mainly during the day and in sunny weather, this counting tool is noticeably less efficient at night or in bad weather (e.g., rain and night can lead to a visible alteration of the image due to reverberations, drop in resolution, motion blur, etc.). The challenge to be faced is therefore multiple: how to detect these new scenarios in real time, carry out a sensitivity analysis of the system in the face of these disturbances, propose a tool guaranteeing the proper functioning of the system in this context, collect, annotate and increase the quantity of this new data for subsequent training?

TO REMEMBER

Advantage of NeuralDE

Data preprocessing, consisting of eliminating raindrops and possibly snowflakes and transforming images from night to day, is what allowed the system to process, without additional training, this data as in normal learning conditions.

The objective of the **industrial application** repository is to present additional methods that are not integrated in the library as being Use-Case specific, but which are constitutive part of the NeuralDE component as a whole. We present them in the context of the Air Liquide Cylinder Counting Use-Case, introducing them gradually while we highlight a methodology on how to concatenate all test-time adaptation methods which are pertinent to this Use-Case. The approach consists on decomposing the test-time adaptation objective into individual domain shift problems, arguing on the root phenomenon causing the shift and systematically proposing a domain shift inversion strategy for each decomposed problem. Then, we gradually recompose the overall test-time adaptation objective as an ordered concatenation of the above-mentioned shift inversion solutions we found. At each step, we evaluate the impact that each individual or composed strategy has on the target model's performance and assess to what extent the original test-time adaptation objective has been met.

In the next subsection we report a table corresponding to the added value of concatenating multiple domain adaptation basic pieces into the NeuralDE pipeline. In particular, one can verify that the robustness gain obtained from concatenating different adaptation modules is better than performance obtained by simply summing up different adaptation modules independently.

E.3 Table 2.

	Mean Error (Misc. bottles)	Relative Error (%)	Error decrease (%)
Baseline	135.3	28.7	-
ResolutionEnhancer	94.8	18.7	10%
NightImageEnhancer	125	26.0	2.7%
ResolutionEnhancer + NightImageEnhancer	93.9	14.6	14.1%

E.4 Secondary Use-Case Demonstrator

The objective of the **NeuralIDE Renault** repository is to address a secondary use-case inspection and application of the NeuralIDE component. The objective of this document is to provide applicability features of the component in other scenarios and their limits and perspectives. We chose the Use-Case Welding Inspection from Renault and adversarial purification as a particular method of interest. We chose this method because it provides a peculiar behavior of the widely known diffusion-based adversarial purification technique which has not been addressed in academic research papers. This behavior can be understood as a failure mode which annihilates the adversarial mask on images but also shifts all images onto one of the prediction classes. This sheds light of the importance of the NeuralIDE methodology to successfully craft good test-time objectives as data augmenting the under-represented class needs to be flagged by the user for improving the calibration of the diffusion auxiliary model. Guidance and class-conditioning mechanisms might also be interesting add-ins to be developed in the future.

F. Conclusions

In its current version, NeuralDE provides a unified way of doing test-time adaptation through input pre-processing, to be used at test time and without the need of retraining or finetuning a pretrained model. It handles three generic types of adaptations:

- To unseen adversarial examples through adversarial purification
- To common corruptions through image restoration
- To Distribution Shifts though Distribution Shift Inversion

As was illustrated in the notebook supports of this guideline, some of these tasks may be computationally expensive at the moment. For instance, adversarial purification is done through diffusion models, which may need a lot of GPU power to be effective. With the rise of new technologies such as Wavelet Diffusion Models for Image Restoration, we have the perspective of making such models more efficient and scalable.

A constitutive limitation of this component is that it is mainly useful when we want to *remove* content (such as adversarial masks, rain drops, snowflakes, etc.). As such, it does not handle tasks in which the main objective is to *add* content (such as object insertion).

Limitations and perspectives

In the near-to-mid future, our perspective is to develop an efficient and comprehensive finetuning scheme for Diffusion-based adversarial purification so that it can handle fine-grained features & imbalanced class sets without distorting them. At the Common Corruption level, we proposed NeuralDE to handle what we considered to be the most difficult corruptions (e.g., rain, snow, fog), as these are usually handled through unsupervised domain adaptation with real world data. We aim to extend our adaptation scheme to the remaining of the 15 different standard corruptions. Another perspective is to generalize NeuralDE to other modalities such as audio or video.

Bibliography

[Py, et al. 2023] Sanders, P. (2023). Real-time Weather Monitoring and Desnowification through Image Purification. In *AAAI 2023 Spring Symposium Series*.

Title: Methodological Guideline for NeuralDE

Keywords:

Domain Adaptation, Test-Time Adaptation, Adversarial Examples, Adversarial Machine Learning, Diffusion Models, Image Restauration Models, Domain Shift, Distribution Shift Inversion, Empirical Robustness, Image-to-Image translations, Input Pre-processing.

Our partners