國立陽明交通大學
NATIONAL YANG MING CHIAO TUNG UNIVERSITY

Unseen Defect Image Synthesis with Compositional Conditional Diffusion Model

Advisor:馬清文教授

Reporter:陳品銓

Date:2024/5/24

CONTENTS

Chapter1

Introduction

Chapter2

Related works

Chapter3

Method

Chapter4

Experiments

&

Results

Chapter5

Conclusion

Chapter1

Introduction

Defect detection

Defect detection

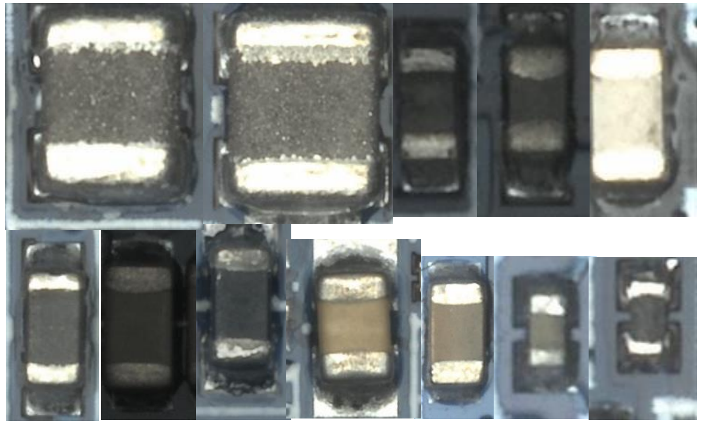
Taiwan's Printed Circuit Board (PCB) industry holds a leading position in global market share. For PCB manufacturers, the yield rate of circuit boards is crucial. Poor yield rates not only increase costs but also damage corporate reputation.

GOAL

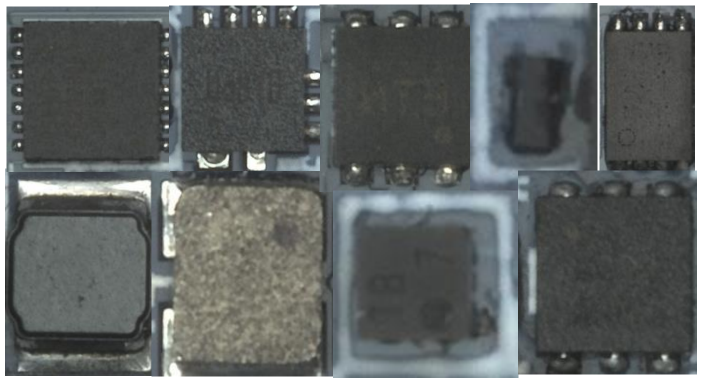
Our goal is to utilize current image generation technology to create unseen defective components

PCB Dataset

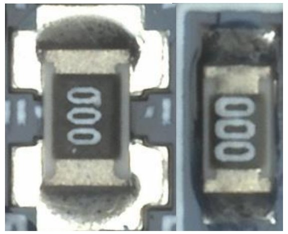
group1



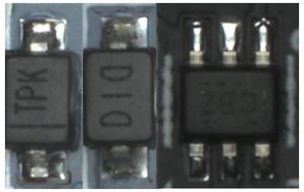
group8



group2

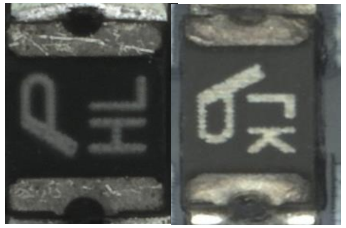


group4



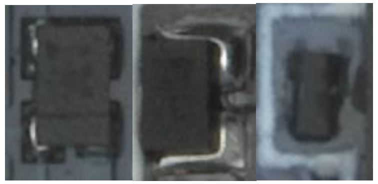
group5

group3



group6

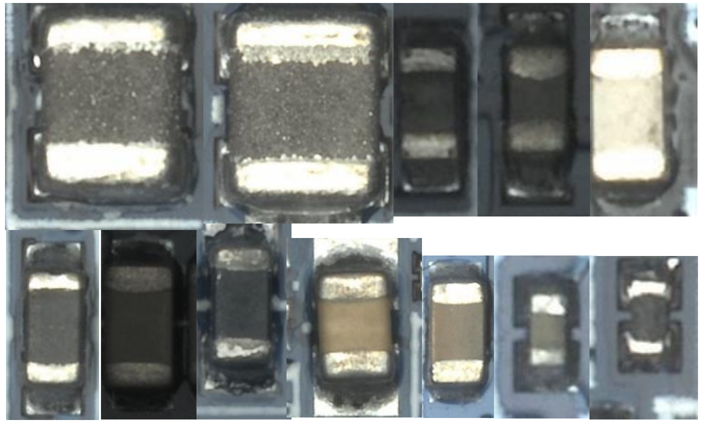
group7



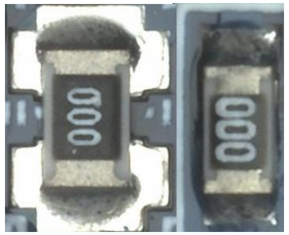
PCB Dataset

	Good	Broke	Shift
Group1	862778	11992	12840
Group2	1084	146	449
Group3	9704	0	88
Group4	9633	488	602
Group5	2777	102	178
Group6	6628	146	76
Group7	18289	295	584
Group8	1279	2	4

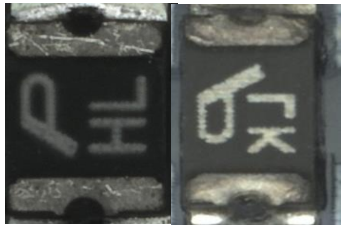
group1



group2

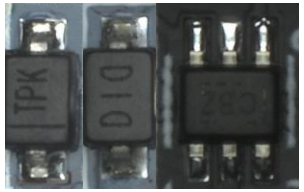


group3

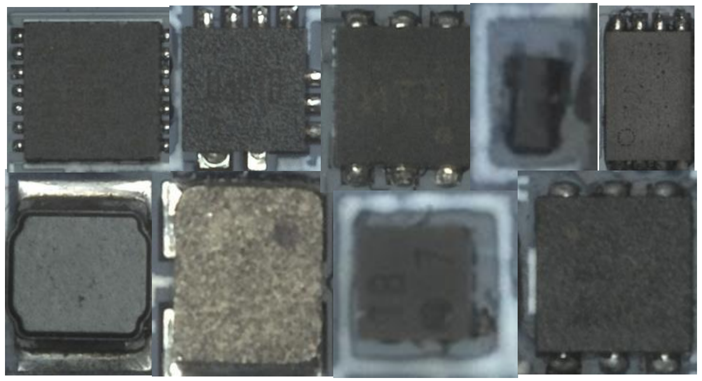


group6

group4

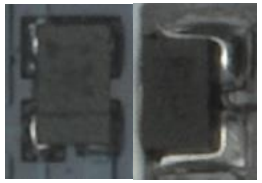


group8

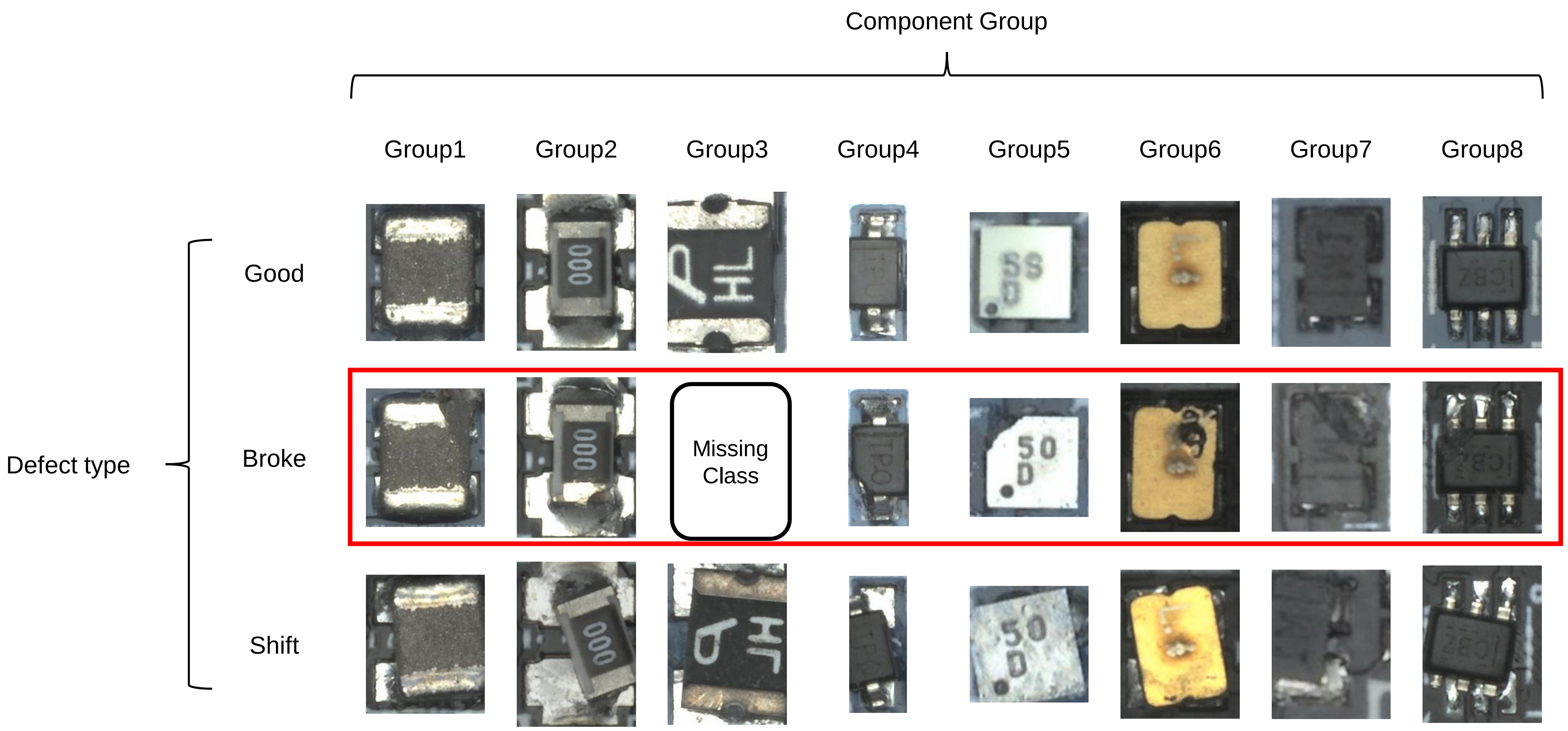


group5

group7



PCB Dataset



Component Group
Group1
Group2
Group3
Group4
Group5
Group6
Group7
Group8
Defect type
Good
Broke
Shift
Missing Class

Chapter2

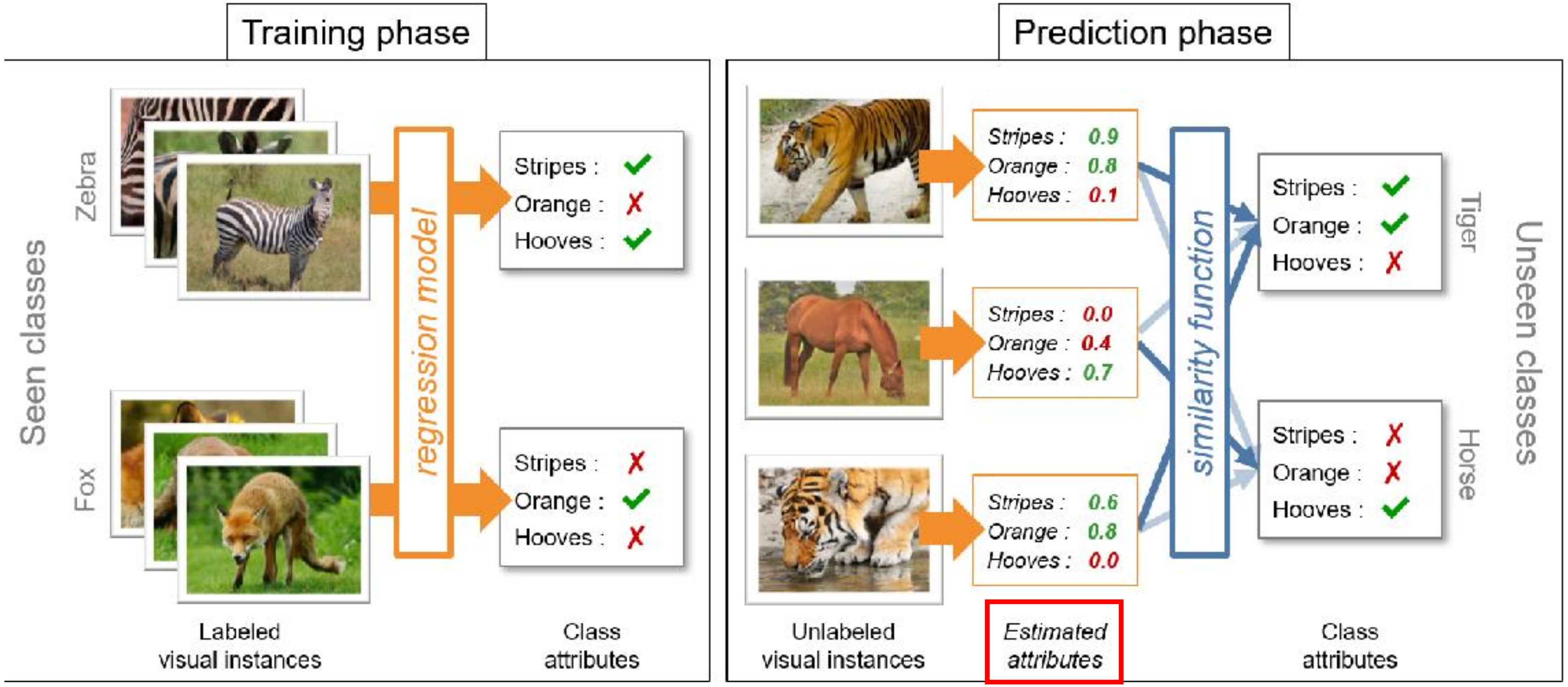
Related works

Compositional Zero-Shot Learning

Compositional Zero-Shot Learning

Compositional Zero-Shot Learning (CZSL) is a computer vision task in which the goal is to recognize unseen compositions fromed from seen state and object during training.

Zero-Shot Learning



Zero-shot learning is a technique that enables pre-trained models to predict class labels of previously unknown data

Image Generation

Image Generation

VAE

Auto-Encoding Variational Bayes

Diederik P. Kingma
Machine Learning Group
Universiteit van Amsterdam
dpkingma@gmail.com

Max Welling
Machine Learning Group
Universiteit van Amsterdam
welling.max@gmail.com

Abstract

How can we perform efficient inference and learning in directed probabilistic models, in the presence of continuous latent variables with intractable posterior distributions, and large datasets? We introduce a stochastic variational inference and learning algorithm that scales to large datasets and, under some mild differentiability conditions, even works in the intractable case. Our contributions are two-fold. First, we show that a reparameterization of the variational lower bound yields a lower bound estimator that can be straightforwardly optimized using standard stochastic gradient methods. Second, we show that for i.i.d. datasets with continuous latent variables per datapoint, posterior inference can be made especially efficient by fitting an approximate inference model (also called a recognition model) to the intractable posterior using the proposed lower bound estimator. Theoretical advantages are reflected in experimental results.

1 Introduction

How can we perform efficient approximate inference and learning with directed probabilistic models whose continuous latent variables and/or parameters have intractable posterior distributions? The variational Bayesian (VB) approach involves the optimization of an approximation to the intractable posterior. Unfortunately, the common mean-field approach requires analytical solutions of expectations w.r.t. the approximate posterior, which are also intractable in the general case. We show how a reparameterization of the variational lower bound yields a simple differentiable unbiased estimator of the lower bound; this SGVB (Stochastic Gradient Variational Bayes) estimator can be used for efficient approximate posterior inference in almost any model with continuous latent variables and/or parameters, and is straightforward to optimize using standard stochastic gradient ascent techniques.

For the case of an i.i.d. dataset and continuous latent variables per datapoint, we propose the Auto-Encoding VB (AEVB) algorithm. In the AEVB algorithm we make inference and learning especially efficient by using the SGVB estimator to optimize a recognition model that allows us to perform very efficient approximate posterior inference using simple ancestral sampling, which in turn allows us to efficiently learn the model parameters, without the need of expensive iterative inference schemes (such as MCMC) per datapoint. The learned approximate posterior inference model can also be used for a host of tasks such as recognition, denoising, representation and visualization purposes. When a neural network is used for the recognition model, we arrive at the *variational auto-encoder*.

2 Method

The strategy in this section can be used to derive a lower bound estimator (a stochastic objective function) for a variety of directed graphical models with continuous latent variables. We will restrict ourselves here to the common case where we have an i.i.d. dataset with latent variables per datapoint, and where we like to perform maximum likelihood (ML) or maximum a posteriori (MAP) inference on the (global) parameters, and variational inference on the latent variables. It is, for example,

GAN

Generative Adversarial Nets

Ian J. Goodfellow, Jean Pouget-Abadie,* Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair,† Aaron Courville, Yoshua Bengio‡
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

Abstract

We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.

1 Introduction

The promise of deep learning is to discover rich, hierarchical models [2] that represent probability distributions over the kinds of data encountered in artificial intelligence applications, such as natural images, audio waveforms containing speech, and symbols in natural language corpora. So far, the most striking successes in deep learning have involved discriminative models, usually those that map a high-dimensional, rich sensory input to a class label [14, 22]. These striking successes have primarily been based on the backpropagation and dropout algorithms, using piecewise linear units [19, 9, 10] which have a particularly well-behaved gradient . Deep *generative* models have had less of an impact, due to the difficulty of approximating many intractable probabilistic computations that arise in maximum likelihood estimation and related strategies, and due to difficulty of leveraging the benefits of piecewise linear units in the generative context. We propose a new generative model estimation procedure that sidesteps these difficulties. [1]

In the proposed *adversarial nets* framework, the generative model is pitted against an adversary: a discriminative model that learns to determine whether a sample is from the model distribution or the data distribution. The generative model can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency. Competition in this game drives both teams to improve their methods until the counterfeits are indistiguishable from the genuine articles.

*Jean Pouget-Abadie is visiting Université de Montréal from Ecole Polytechnique.
†Sherjil Ozair is visiting Université de Montréal from Indian Institute of Technology Delhi
‡Yoshua Bengio is a CIFAR Senior Fellow.
[1] All code and hyperparameters available at http://www.github.com/goodfeli/adversarial

VAE

Auto-Encoding Variational Bayes

Diederik P. Kingma
Machine Learning Group
Universiteit van Amsterdam
dpkingma@gmail.com

Max Welling
Machine Learning Group
Universiteit van Amsterdam
welling.max@gmail.com

Abstract

How can we perform efficient inference and learning in directed probabilistic models, in the presence of continuous latent variables with intractable posterior distributions, and large datasets? We introduce a stochastic variational inference and learning algorithm that scales to large datasets and, under some mild differentiability conditions, even works in the intractable case. Our contributions are two-fold. First, we show that a reparameterization of the variational lower bound yields a lower bound estimator that can be straightforwardly optimized using standard stochastic gradient methods. Second, we show that for i.i.d. datasets with continuous latent variables per datapoint, posterior inference can be made especially efficient by fitting an approximate inference model (also called a recognition model) to the intractable posterior using the proposed lower bound estimator. Theoretical advantages are reflected in experimental results.

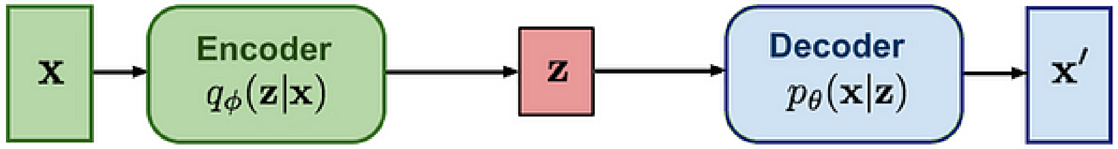
1 Introduction

How can we perform efficient approximate inference and learning with directed probabilistic models whose continuous latent variables and/or parameters have intractable posterior distributions? The variational Bayesian (VB) approach involves the optimization of an approximation to the intractable posterior. Unfortunately, the common mean-field approach requires analytical solutions of expectations w.r.t. the approximate posterior, which are also intractable in the general case. We show how a reparameterization of the variational lower bound yields a simple differentiable unbiased estimator of the lower bound; this SGVB (Stochastic Gradient Variational Bayes) estimator can be used for efficient approximate posterior inference in almost any model with continuous latent variables and/or parameters, and is straightforward to optimize using standard stochastic gradient ascent techniques.

For the case of an i.i.d. dataset and continuous latent variables per datapoint, we propose the Auto-Encoding VB (AEVB) algorithm. In the AEVB algorithm we make inference and learning especially efficient by using the SGVB estimator to optimize a recognition model that allows us to perform very efficient approximate posterior inference using simple ancestral sampling, which in turn allows us to efficiently learn the model parameters, without the need of expensive iterative inference schemes (such as MCMC) per datapoint. The learned approximate posterior inference model can also be used for a host of tasks such as recognition, denoising, representation and visualization purposes. When a neural network is used for the recognition model, we arrive at the *variational auto-encoder*.

2 Method

The strategy in this section can be used to derive a lower bound estimator (a stochastic objective function) for a variety of directed graphical models with continuous latent variables. We will restrict ourselves here to the common case where we have an i.i.d. dataset with latent variables per datapoint, and where we like to perform maximum likelihood (ML) or maximum a posteriori (MAP) inference on the (global) parameters, and variational inference on the latent variables. It is, for example,



- Low-fidelity samples
- Easy to train
- Mode coverage(Diversity)

[19] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” arXiv preprint arXiv:1312.6114, 2013.

GAN

- High-fidelity samples
- Low diversity samples
- Hard to train

Generative Adversarial Nets

Ian J. Goodfellow, Jean Pouget-Abadie*, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair†, Aaron Courville, Yoshua Bengio‡
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

Abstract

We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.

1 Introduction

The promise of deep learning is to discover rich, hierarchical models [2] that represent probability distributions over the kinds of data encountered in artificial intelligence applications, such as natural images, audio waveforms containing speech, and symbols in natural language corpora. So far, the most striking successes in deep learning have involved discriminative models, usually those that map a high-dimensional, rich sensory input to a class label [14, 22]. These striking successes have primarily been based on the backpropagation and dropout algorithms, using piecewise linear units [19, 9, 10] which have a particularly well-behaved gradient . Deep *generative* models have had less of an impact, due to the difficulty of approximating many intractable probabilistic computations that arise in maximum likelihood estimation and related strategies, and due to difficulty of leveraging the benefits of piecewise linear units in the generative context. We propose a new generative model estimation procedure that sidesteps these difficulties. [1]

In the proposed *adversarial nets* framework, the generative model is pitted against an adversary: a discriminative model that learns to determine whether a sample is from the model distribution or the data distribution. The generative model can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency. Competition in this game drives both teams to improve their methods until the counterfeits are indistiguishable from the genuine articles.

* Jean Pouget-Abadie is visiting Université de Montréal from Ecole Polytechnique.
† Sherjil Ozair is visiting Université de Montréal from Indian Institute of Technology Delhi
‡ Yoshua Bengio is a CIFAR Senior Fellow.
[1] All code and hyperparameters available at `http://www.github.com/goodfeli/adversarial`

[18] I.Goodfellow,J.Pouget-Abadie, M.Mirza,B.Xu,D.Warde-Farley,S.Ozair,A.Courville, and Y. Bengio, "Generative adversarial networks," Communications of the ACM, vol. 63, no. 11, pp. 139–144, 2020.

Diffusion Models

Diffusion Models

Deep Unsupervised Learning using Nonequilibrium Thermodynamics

Jascha Sohl-Dickstein JASCHA@STANFORD.EDU
Stanford University

Eric A. Weiss EAWEISS@BERKELEY.EDU
University of California, Berkeley

Niru Maheswaranathan NIRUM@STANFORD.EDU
Stanford University

Surya Ganguli SGANGULI@STANFORD.EDU
Stanford University

Abstract

A central problem in machine learning involves modeling complex data-sets using highly flexible families of probability distributions in which learning, sampling, inference, and evaluation are still analytically or computationally tractable. Here, we develop an approach that simultaneously achieves both flexibility and tractability. The essential idea, inspired by non-equilibrium statistical physics, is to systematically and slowly destroy structure in a data distribution through an iterative forward diffusion process. We then learn a reverse diffusion process that restores structure in data, yielding a highly flexible and tractable generative model of the data. This approach allows us to rapidly learn, sample from, and evaluate probabilities in deep generative models with thousands of layers or time steps, as well as to compute conditional and posterior probabilities under the learned model. We additionally release an open source reference implementation of the algorithm.

1. Introduction

Historically, probabilistic models suffer from a tradeoff between two conflicting objectives: *tractability* and *flexibility*. Models that are *tractable* can be analytically evaluated and easily fit to data (e.g. a Gaussian or Laplace). However, these models are unable to aptly describe structure in rich datasets. On the other hand, models that are *flexible* can be molded to fit structure in arbitrary data. For example, we can define models in terms of any (non-negative) function $\phi(\mathbf{x})$ yielding the flexible distribution $p(\mathbf{x}) = \frac{\phi(\mathbf{x})}{Z}$, where Z is a normalization constant. However, computing this normalization constant is generally intractable. Evaluating, training, or drawing samples from such flexible models typically requires a very expensive Monte Carlo process.

A variety of analytic approximations exist which ameliorate, but do not remove, this tradeoff–for instance mean field theory and its expansions (T, 1982; Tanaka, 1998), variational Bayes (Jordan et al., 1999), contrastive divergence (Welling & Hinton, 2002; Hinton, 2002), minimum probability flow (Sohl-Dickstein et al., 2011b;a), minimum KL contraction (Lyu, 2011), proper scoring rules (Gneiting & Raftery, 2007; Parry et al., 2012), score matching (Hyvärinen, 2005), pseudolikelihood (Besag, 1975), loopy belief propagation (Murphy et al., 1999), and many, many more. Non-parametric methods (Gershman & Blei, 2012) can also be very effective[1].

1.1. Diffusion probabilistic models

We present a novel way to define probabilistic models that allows:

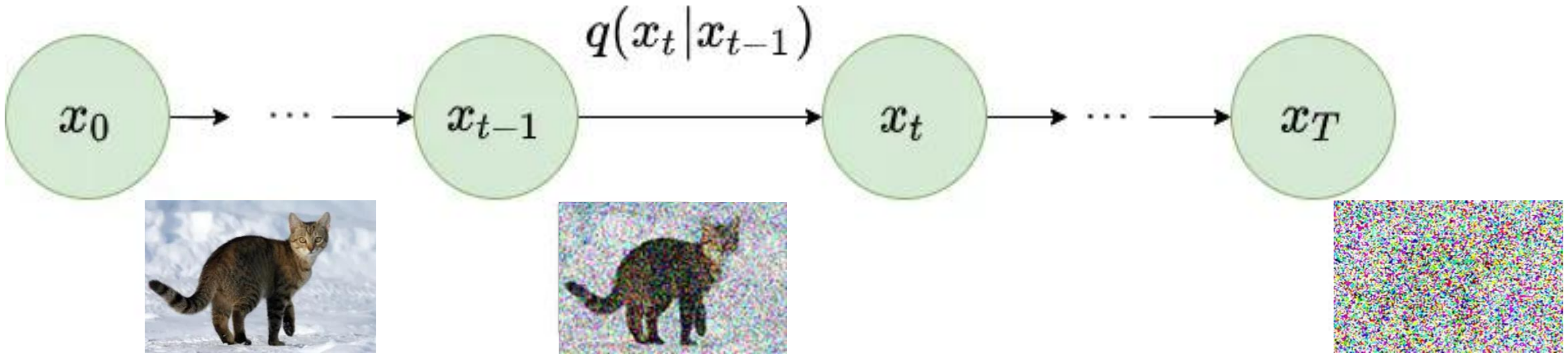
1. extreme flexibility in model structure,
2. exact sampling,

Proceedings of the 32nd *International Conference on Machine Learning*, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

[1] Non-parametric methods can be seen as transitioning smoothly between tractable and flexible models. For instance, a non-parametric Gaussian mixture model will represent a small amount of data using a single Gaussian, but may represent infinite data as a mixture of an infinite number of Gaussians.

[29] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” pp. 2256–2265, 2015

Diffusion Models



Forward Diffusion Process:

Output

Mean μ_t

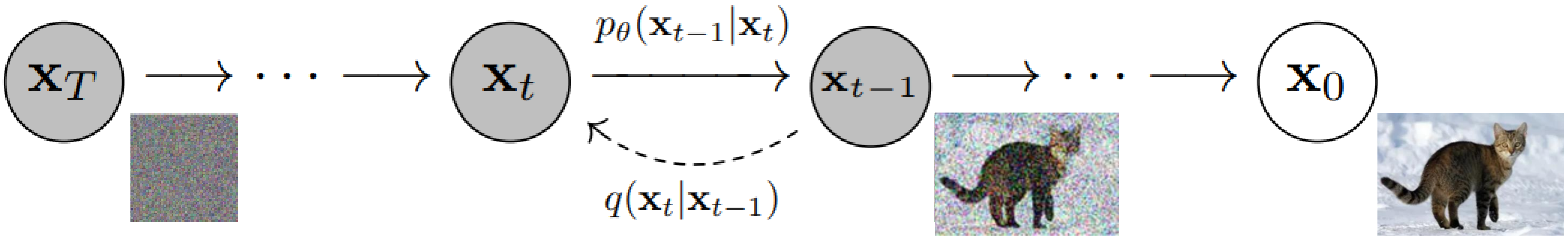
$$\boxed{q(x_t|x_{t-1})} = N(\boxed{x_t}; \boxed{\sqrt{1-\beta_t}x_{t-1}}, \boxed{\beta_t I})$$

Distribution of the noised images

VarianceΣ_t

[20] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," Advances in neural information processing systems, vol. 33, pp. 6840–6851, 2020

Diffusion Models



Reverse Diffusion Process:

Reverse Diffusion

Forward Diffusion

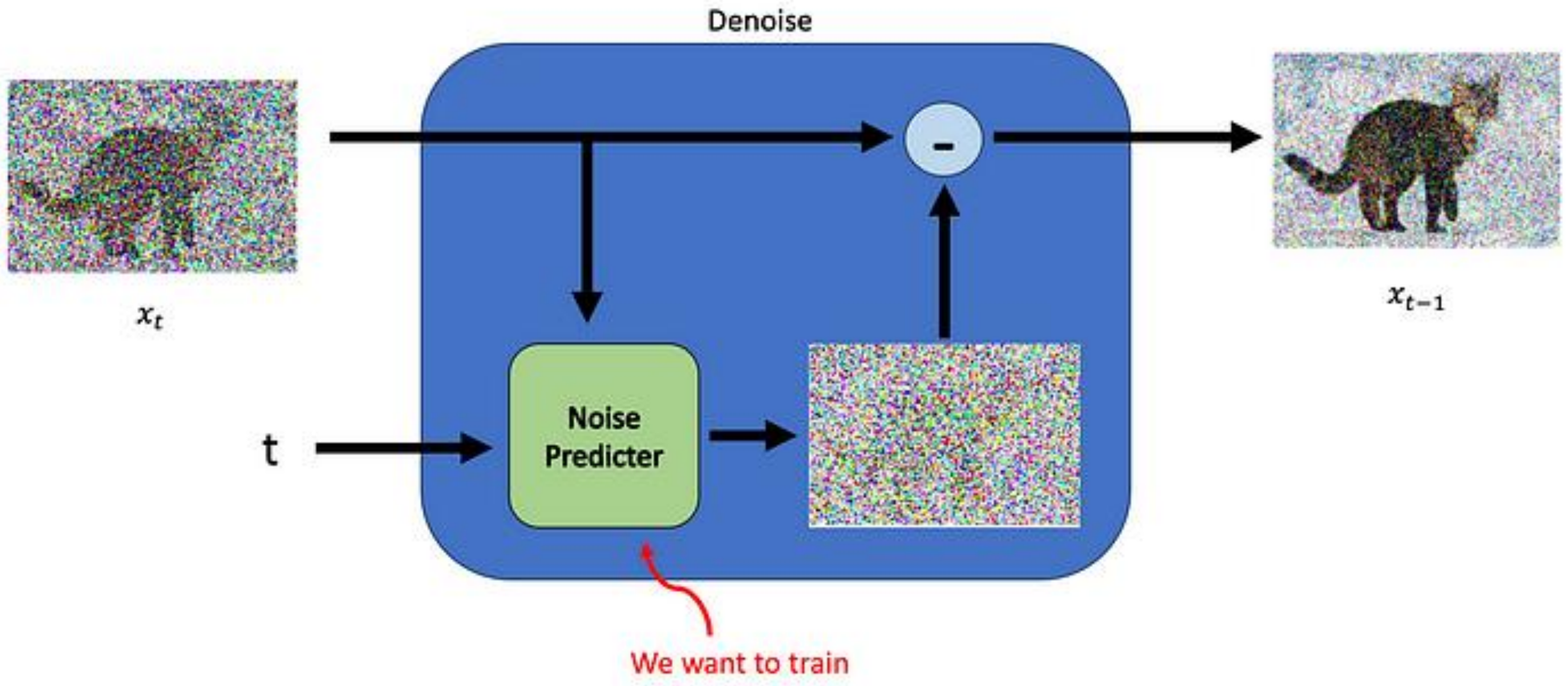
Mean μ_t

$$p_\theta(x_{t-1}|x_t) = N(x_{t-1}; \boxed{\mu_\theta(x_t, t)}, \boxed{\Sigma_\theta(x_t, t)})$$

VarianceΣ_t

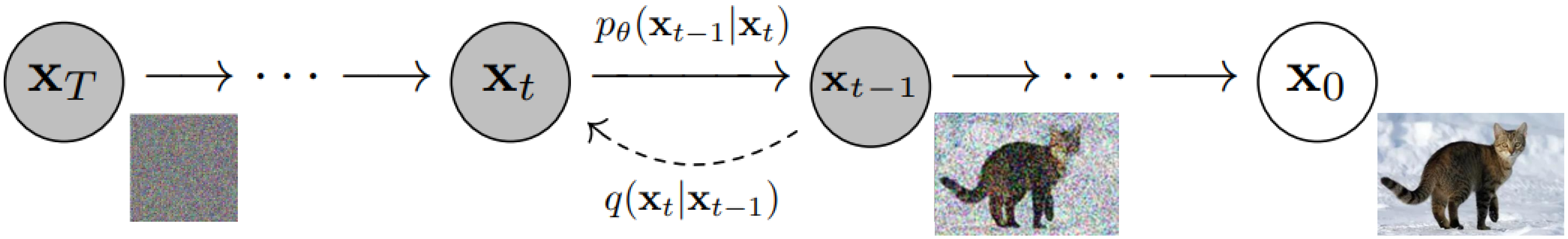
[20] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” Advances in neural information processing systems, vol. 33, pp. 6840–6851, 2020

Diffusion Models



[20] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” Advances in neural information processing systems, vol. 33, pp. 6840–6851, 2020

Diffusion Models



Reverse Diffusion Process:

→ Reverse Diffusion

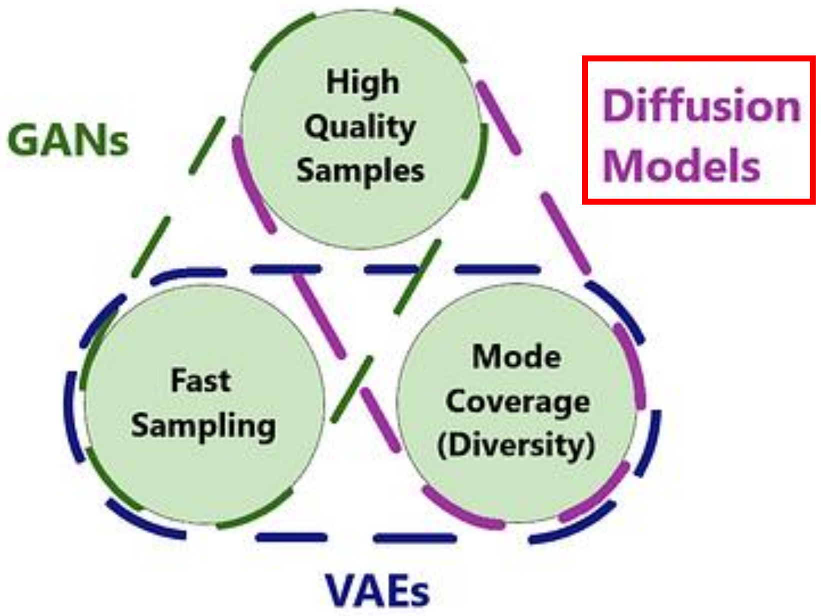
⇠ Forward Diffusion

Predicted noise

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boxed{\epsilon_\theta(x_t, t)}\right)$$

[20] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” Advances in neural information processing systems, vol. 33, pp. 6840–6851, 2020

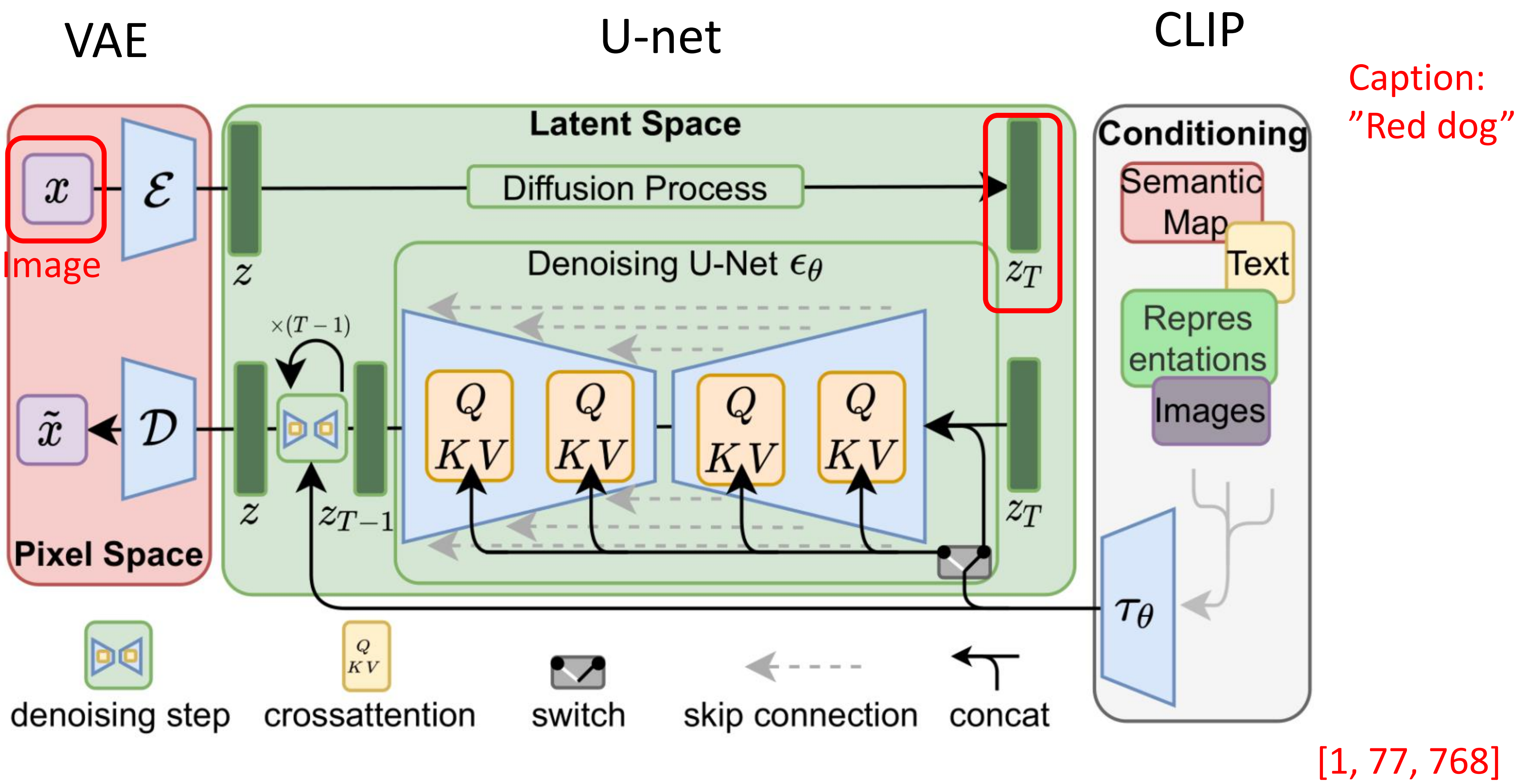
Compare



GANs
High Quality Samples
Diffusion Models
Fast Sampling
Mode Coverage (Diversity)
VAEs

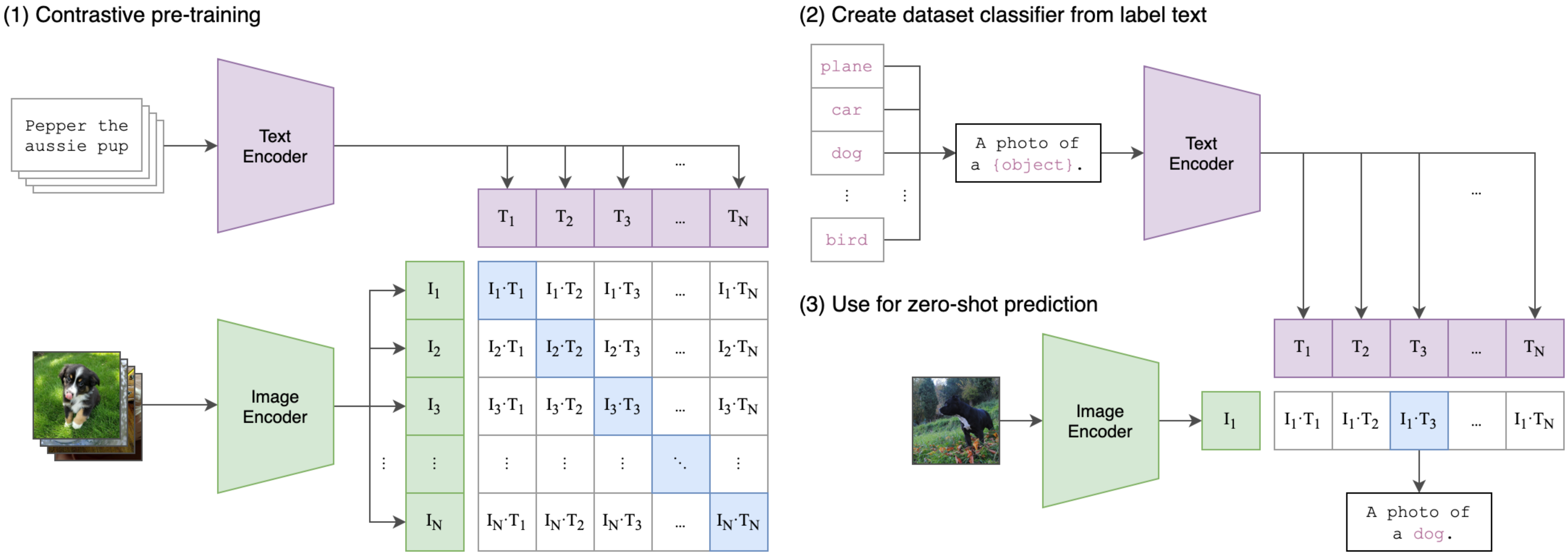
Conditional Diffusion Models

Stable diffusion



[1] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," pp. 10684–10695, 2022

Contrastive Language-Image Pre-Training(CLIP)



[3] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark et al., "Learning transferable visual models from natural language supervision," pp. 8748–8763, 2021.

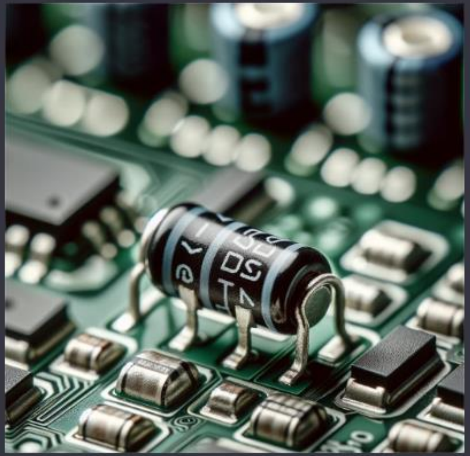
Why not use off-the-shelf Stable Diffusion(e.g. ChatGPT)?

Why not use off the shelf Stable Diffusion?

You

This is an image of SMD component on PCB borad.
Can you generate similar images?

ChatGPT



Here is a generated image of a surface-mounted device (SMD) on a printed circuit board (PCB), similar to the one you provided.

You

What is the prompt of the generated image?

You

What is the prompt of the generated image?

ChatGPT

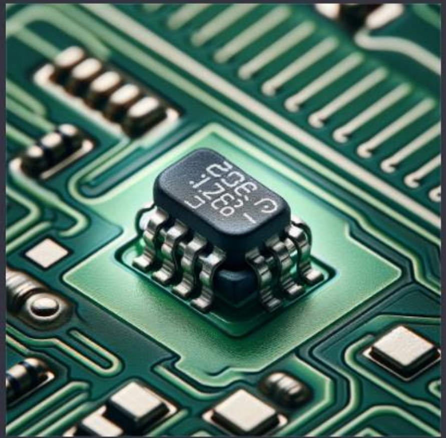
The prompt used for the generated image was:

"A close-up image of a surface-mounted device (SMD) component on a green printed circuit board (PCB). The focus is on a singular SMD resistor with a black body and white printed code on top, seated amongst other electronic components such as capacitors and integrated circuits. The solder joints are clearly visible, attaching the components securely to the copper traces on the PCB, which is populated with various SMD components showcasing intricate electronic design."

You

The image I want is taken from top view and only one SMD component

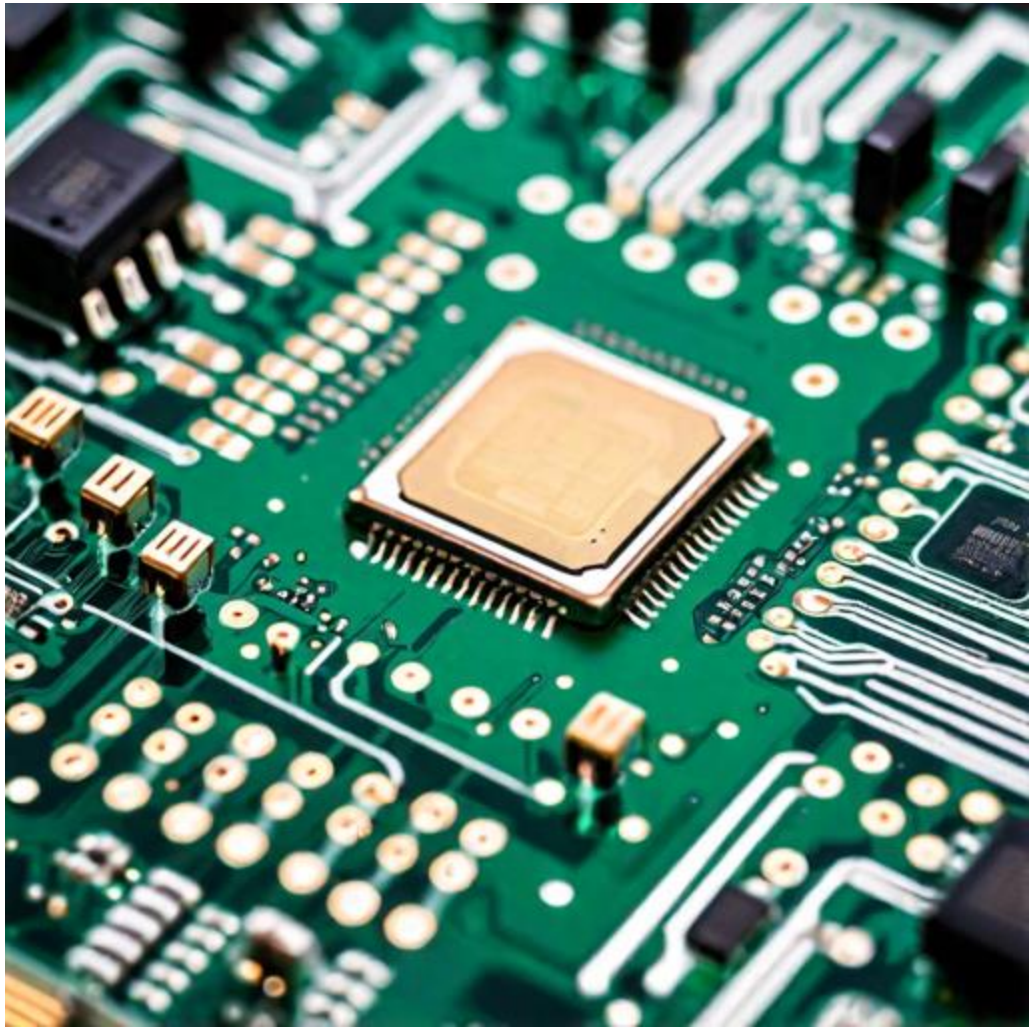
ChatGPT



Here is the updated image based on your specifications, taken from a top view featuring a single SMD component on a PCB.

Why not use off the shelf Stable Diffusion?

"A close up image of a surface mounted device SMD component on a printed circuit board."

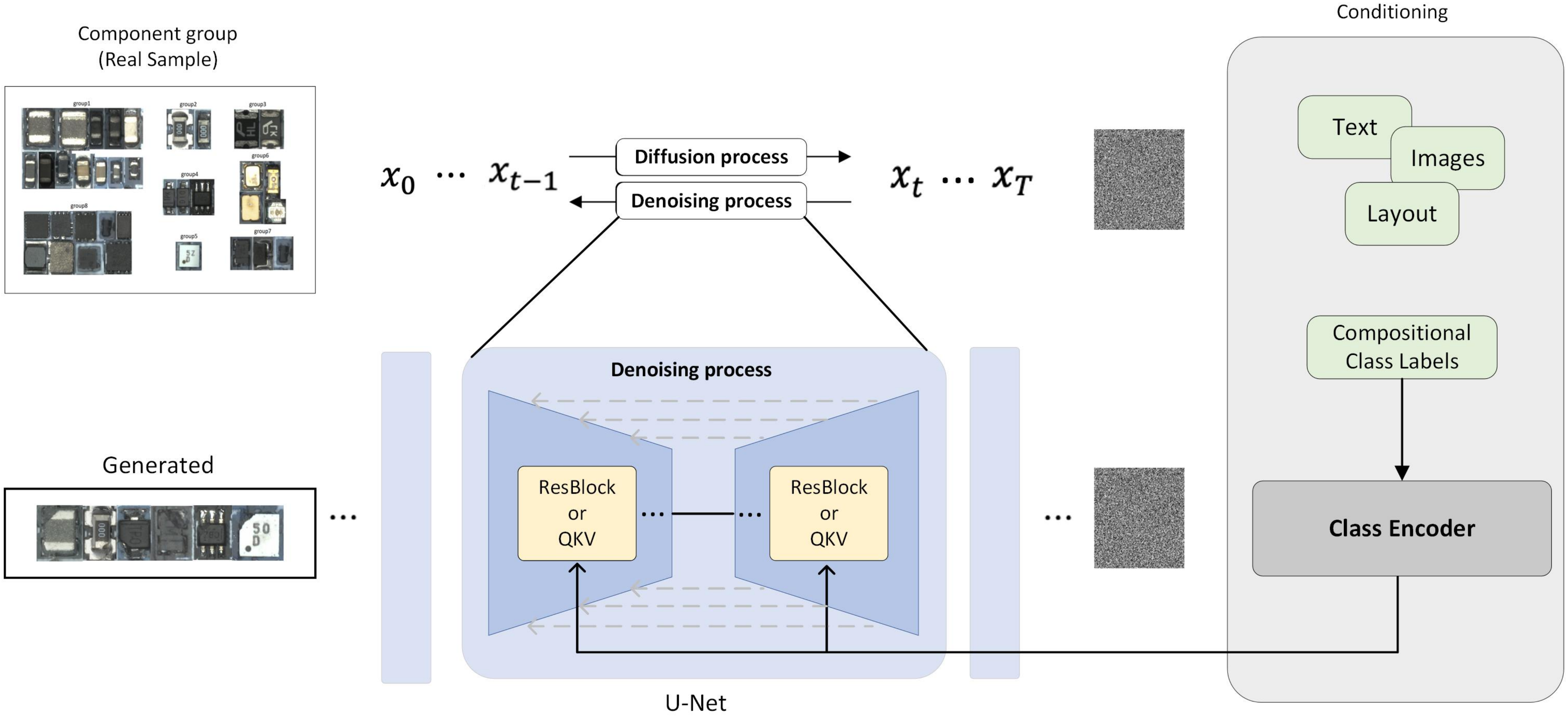


Chapter3

Method

Compositional Conditional Diffusion Models

Compositional Conditional Diffusion Models



Forward Diffusion Process:

$$q(x_t|x_{t-1}) = N\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I\right)$$

Reverse Diffusion Process:

$$p_\theta(x_{t-1}|x_t) = N\left(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right)$$

Forward Diffusion Process:

$$q(x_t|x_{t-1}, c) = N\left(x_t|c; \sqrt{1-\beta_t}x_{t-1}|c, \beta_t I\right)$$

Reverse Diffusion Process:

$$p_\theta(x_{t-1}|x_t, c) = N\left(x_{t-1}; \mu_\theta(x_t, t, c), \Sigma_\theta(x_t, t, c)\right)$$

[26] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” arXiv preprint arXiv:2207.12598, 2022

Forward Diffusion Process:

$$q(x_t|x_{t-1}, c) = N\left(x_t|c; \sqrt{1-\beta_t}x_{t-1}|c, \beta_t I\right)$$

Reverse Diffusion Process:

$$p_\theta(x_{t-1}|x_t, c) = N\left(x_{t-1}; \mu_\theta(x_t, t, c), \Sigma_\theta(x_t, t, c)\right)$$

$$c\ such\ as\ text, semantic\ maps, etc$$

$$In\ our\ case, c = (c_1,\ c_2,\ c_3, \ldots)\ represents\ a\ composite\ condition\ encoding$$

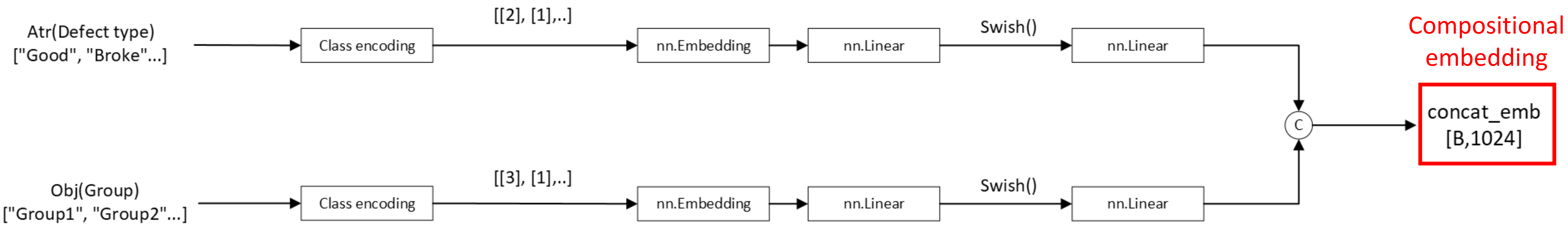
[26] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” arXiv preprint arXiv:2207.12598, 2022

$$\textcolor{red}{c}\ such\ as\ text, semantic\ maps, etc$$

$$In\ our\ case, c = (c_1, c_2, c_3, \ldots)\ represents\ a\ composite\ condition\ encoding$$

$$c_{atr} = Proj(Emb(Encoder(atr)))$$

$$c_{obj} = Proj(Emb(Encoder(obj)))$$



[26] J. Ho and T. Salimans, "Classifier-free diffusion guidance," arXiv preprint arXiv:2207.12598, 2022

Reverse Diffusion Process:

$$p_\theta(x_{t-1}|x_t, c) = N\left(x_{t-1}; \mu_\theta(x_t, t, c), \Sigma_\theta(x_t, t, c)\right)$$

$$\mu_\theta(x_t, t, c) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \overbrace{\boxed{\epsilon_\theta(x_t, t, c)}}^{\text{Predicted noise}}\right)$$

[26] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” arXiv preprint arXiv:2207.12598, 2022

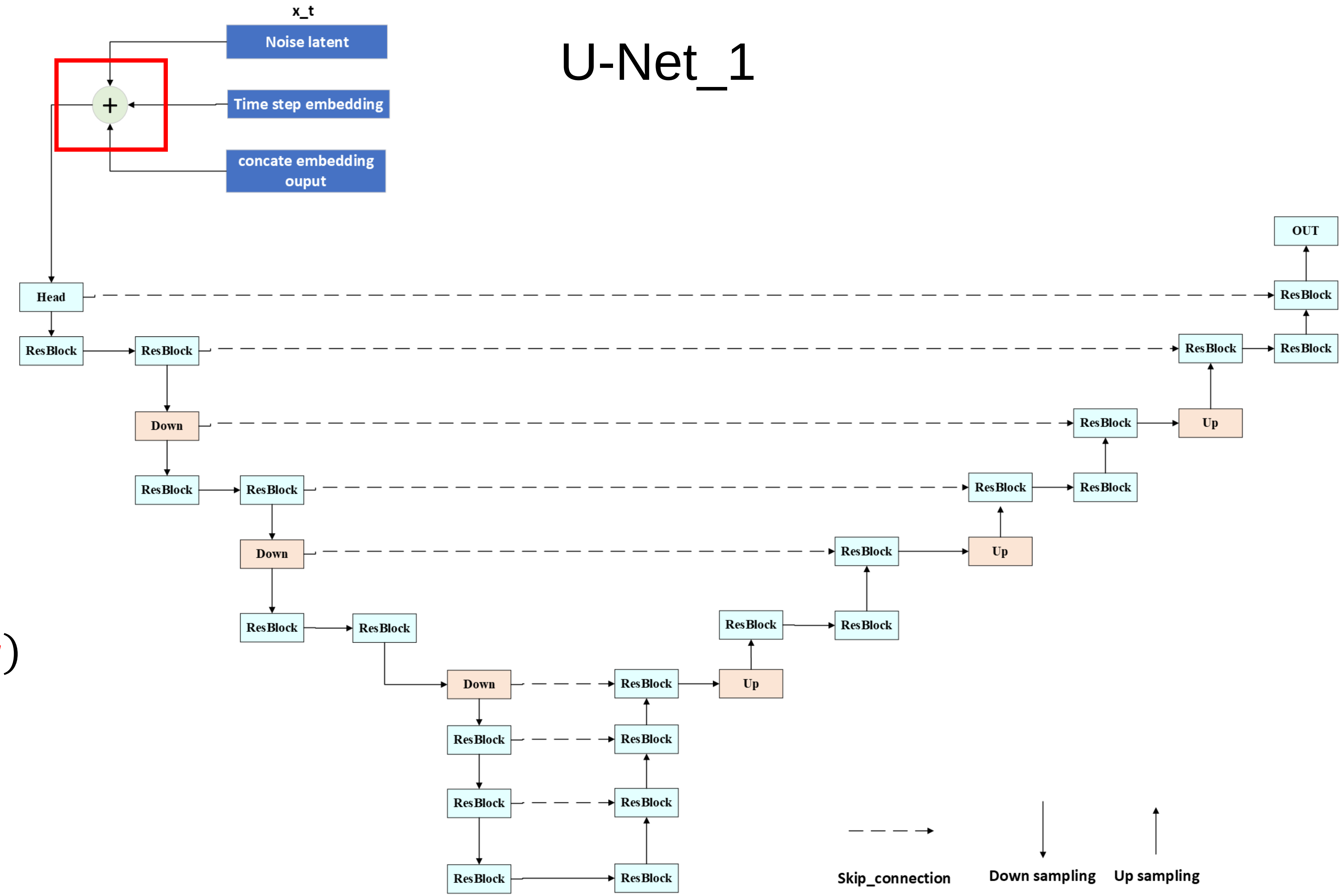
U-Net

U-Net

	U-Net_1	U-Net_2
ResBlock number	2	2
Condition embedding	Fix / learnable	Fix
Condition methods	Classifier free guidance	
Add condition embedding	AddGN 1(c, t), AdaGN 1(c, t)	QKV(h, cMLIP(c))
Add Time embedding	AddGN 1(c, t), AdaGN 1(c, t)	AddGN 2(t)

U-Net_1

U-Net_1



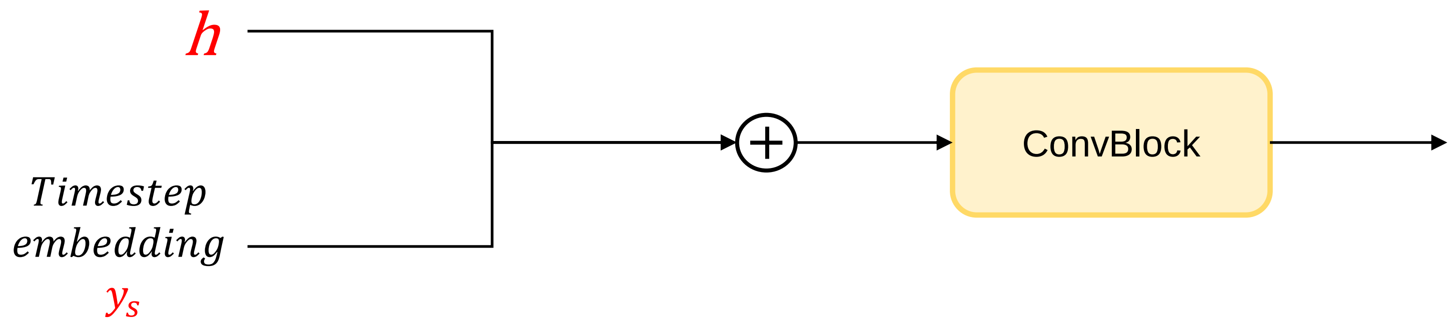
$$\epsilon_\theta(x_t, t, c)$$

U-Net_1

We define this layer as addition GroupNorm(AddGN)

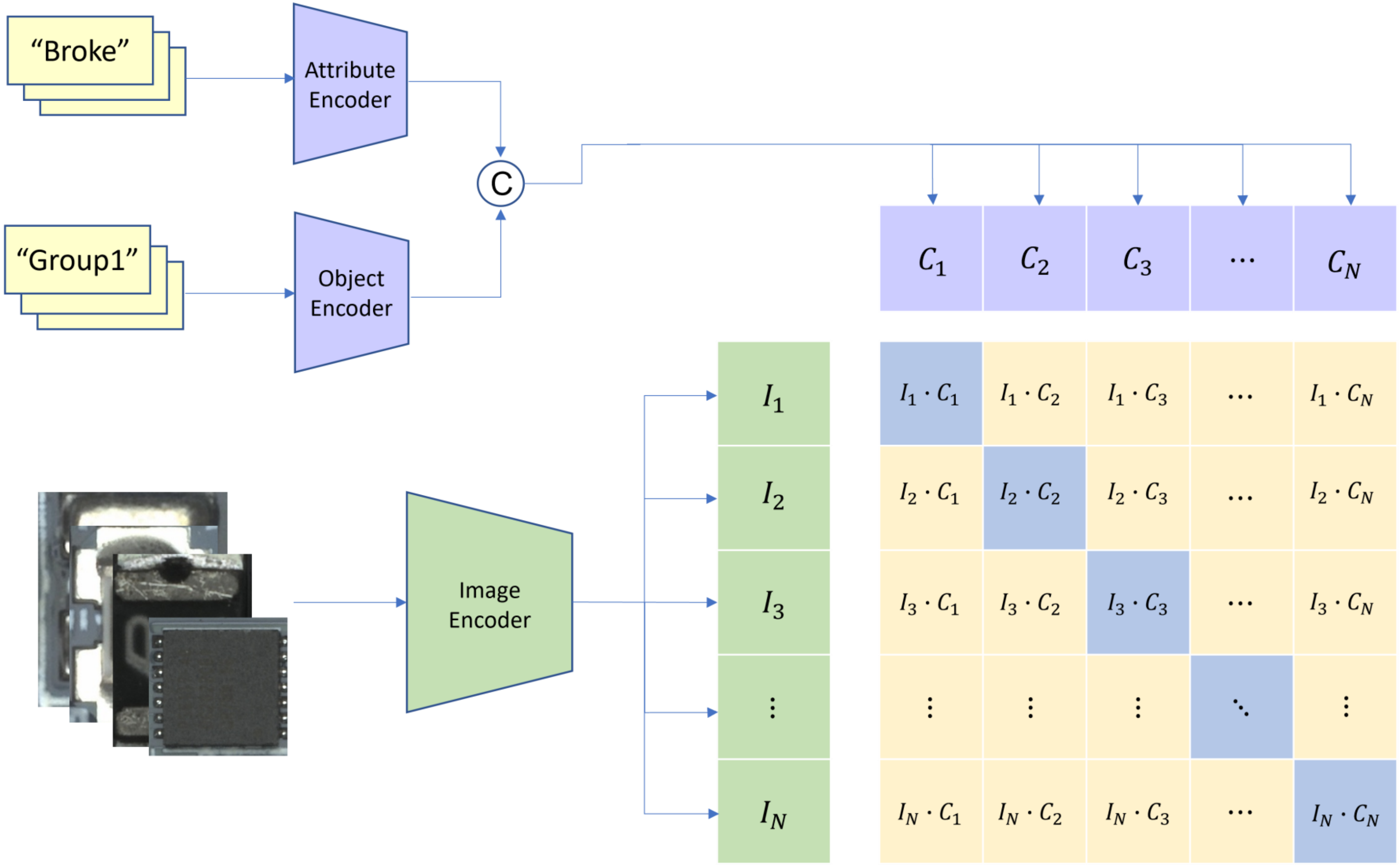
$$AddGN(h, y_s) = y_s + GroupNorm(h)$$

where h is the intermediate activations of the resblock following the first convolution y_s is a linear projection of the timestep



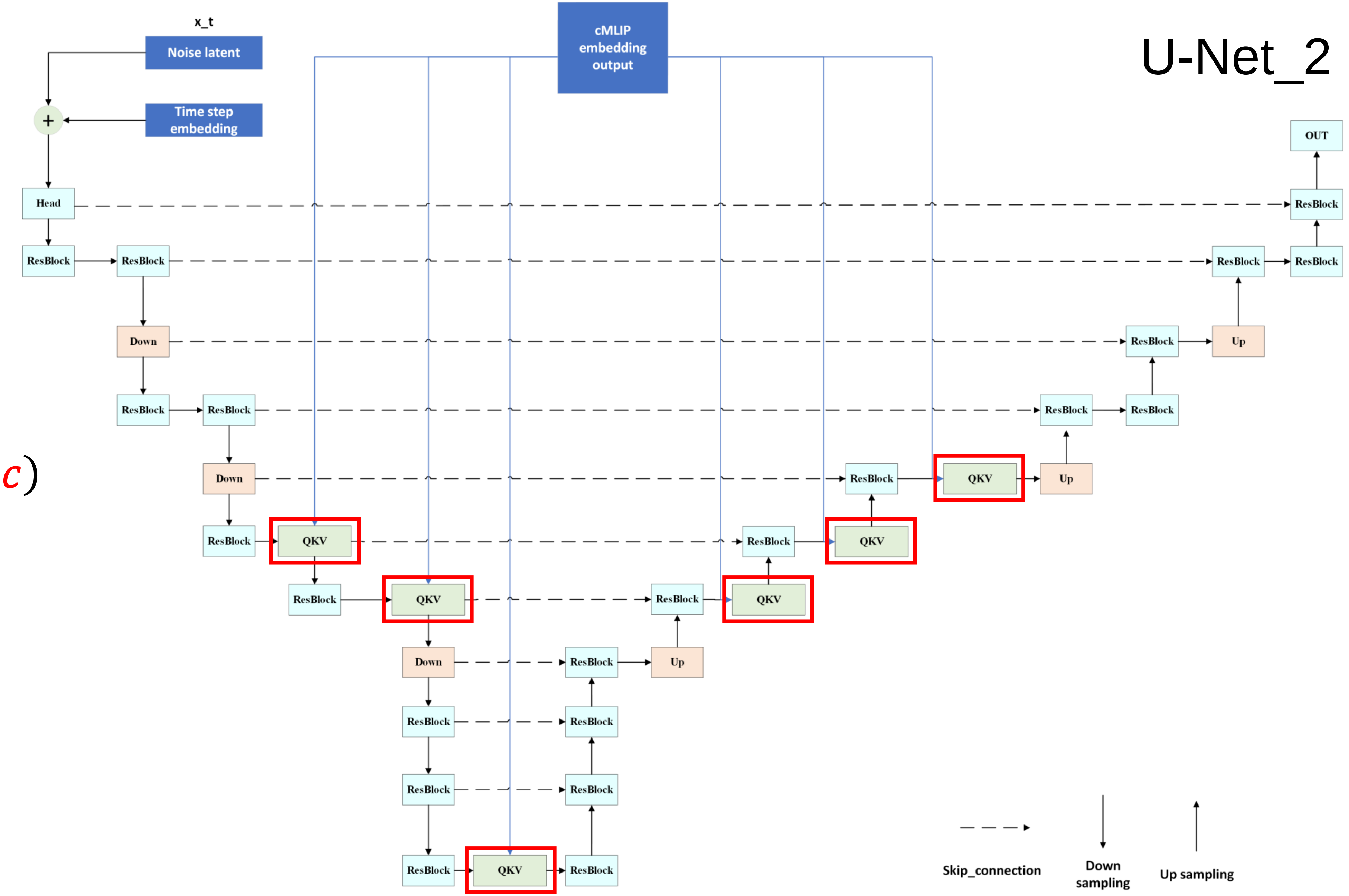
U-Net_2

Contrastive Multi-Label Image Pre-Training(cMLIP)



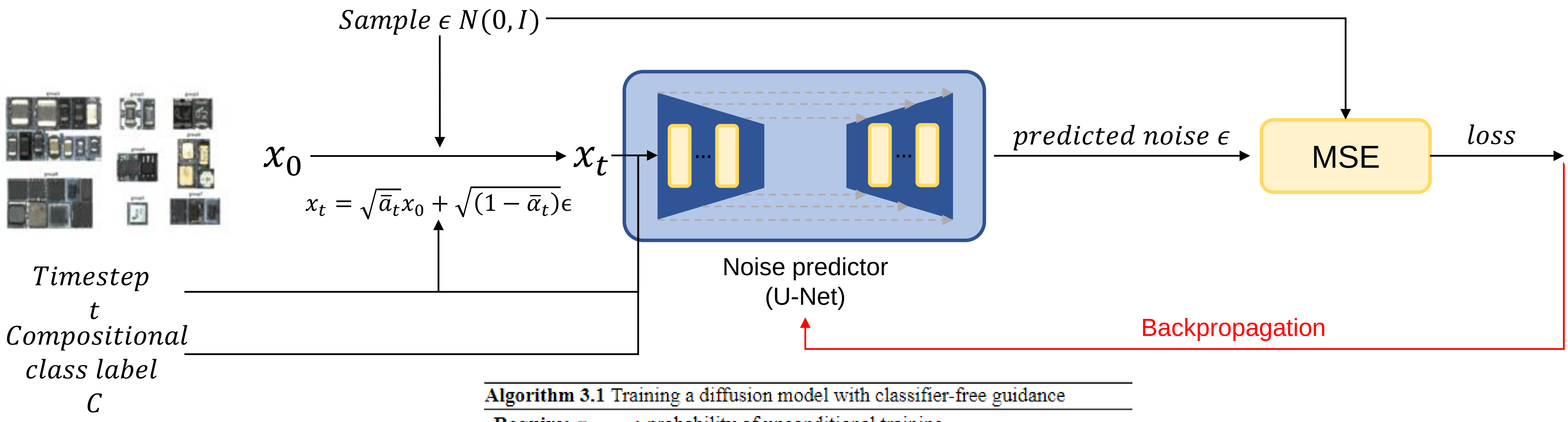
U-Net_2

$$\epsilon_\theta(x_t, t, c)$$



Algorithm

Algorithm



Algorithm 3.1 Training a diffusion model with classifier-free guidance

Require: p_{uncond}: probability of unconditional training

repeat

  $(x_0, c) \sim q(x_0, c)$

  $c \leftarrow \emptyset$ with probability p_{uncond}

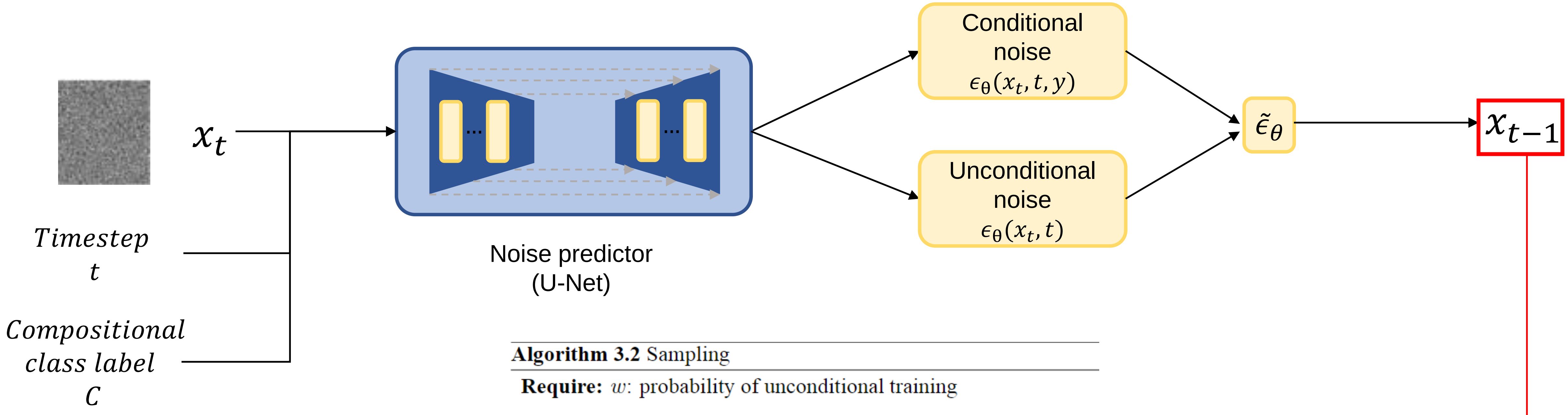
  $\epsilon \sim \mathcal{N}(0, I)$

  $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{(1-\bar{\alpha}_t)}\epsilon$

  Take gradient descent step on $\nabla_\theta || \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{(1-\bar{\alpha}_t)}\epsilon, c, t)) - \epsilon||^2$

Until converged

Algorithm



Algorithm 3.2 Sampling

Require: w: probability of unconditional training

$x_T \sim \mathcal{N}(0, I)$, $c \sim p(c)$

for t = T, …, 1 **do**

$\quad z \sim \mathcal{N}(0, I)$if $t > 1$,else z = 0

$\quad \tilde{\epsilon}_\theta = (w+1)\epsilon_\theta(x_t, c, t) - w\epsilon_\theta(x_t, t)$

$\quad \boxed{x_{x-1} = \sqrt{\alpha_{t-1}}(\frac{x_t - \sqrt{1-\alpha_t}\tilde{\epsilon}_\theta}{\sqrt{\alpha_t}}) + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \tilde{\epsilon}_\theta + \sigma_t \epsilon_t}$

end for

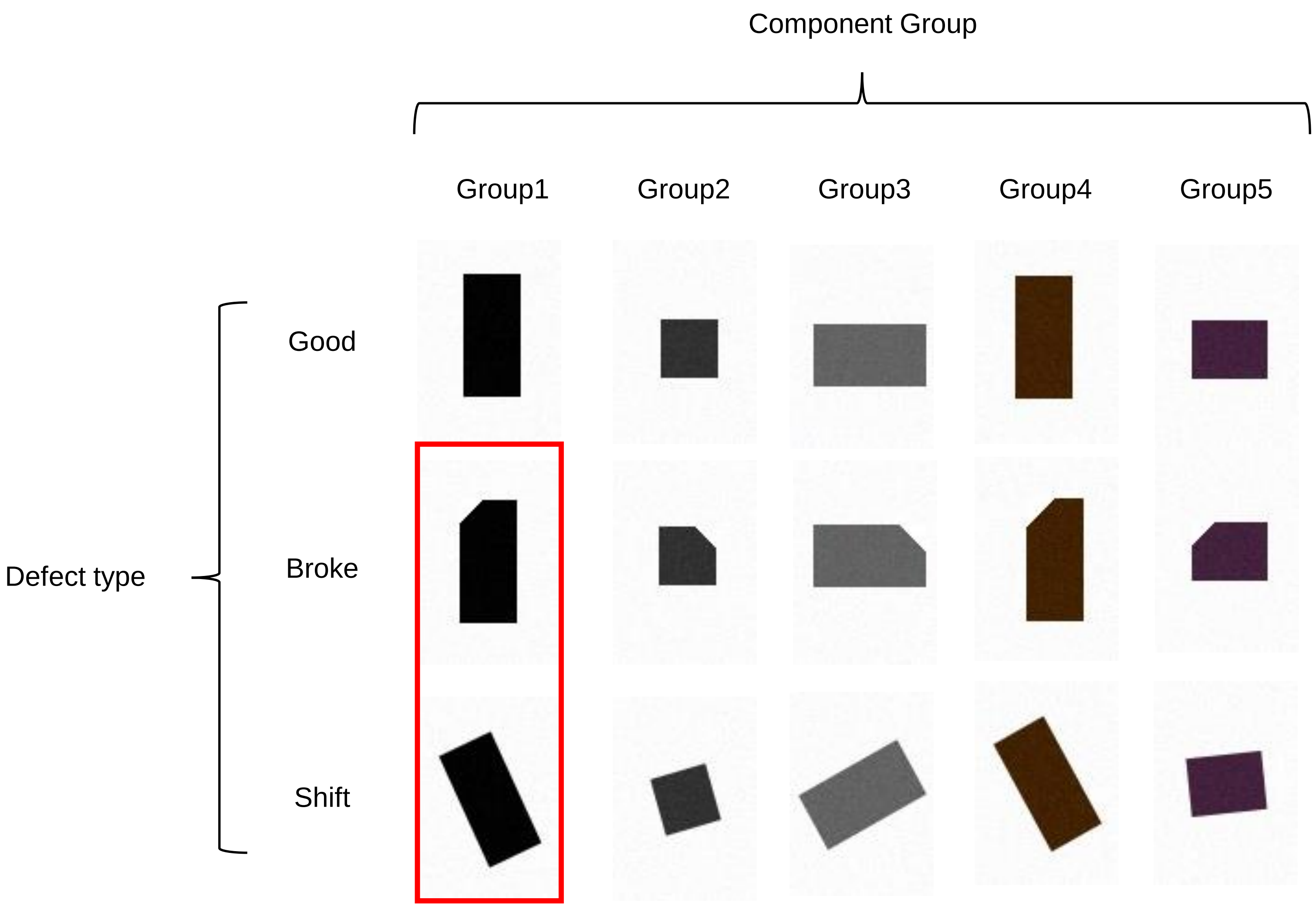
return x_0

Chapter4

Experiments
&
Results

Toy Dataset

Toy Dataset



Component Group
Group1
Group2
Group3
Group4
Group5
Defect type
Good
Broke
Shift

Toy Dataset

cMLIP	
Epoch	10
Batch size	16
Learning rate	1e-4
Dropout	0.15
Weight Decay	1e-4
Embedding dimension	512
Projection dimension	256

Toy Dataset

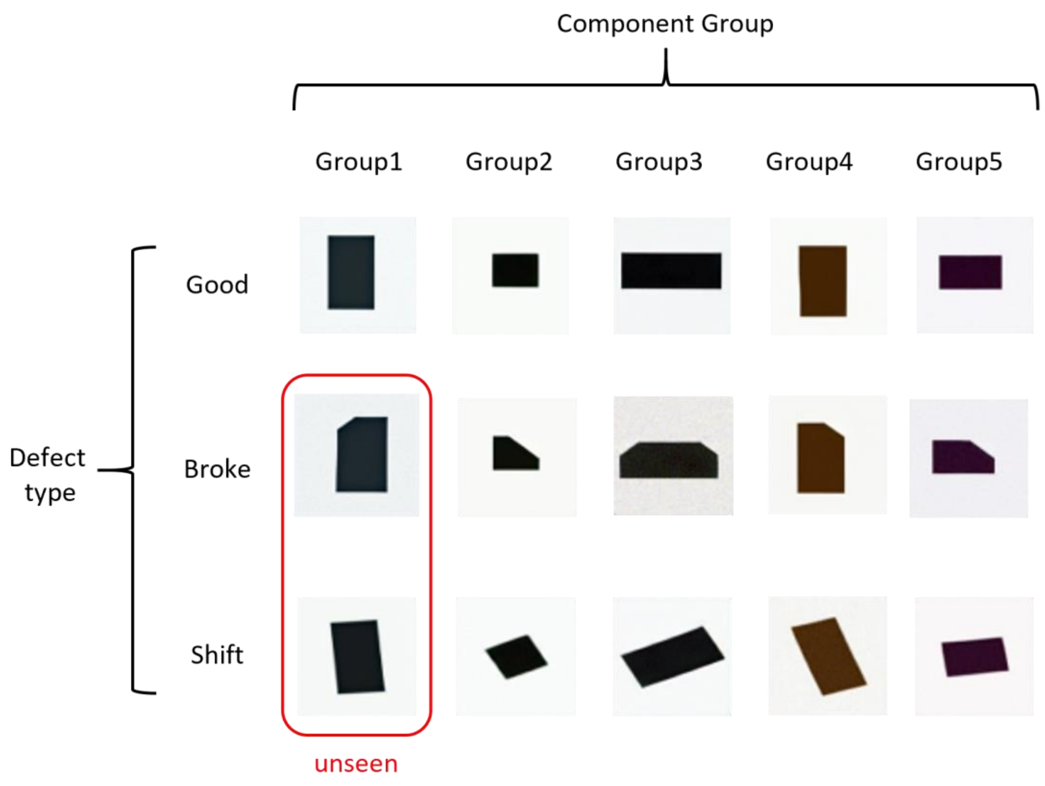
cMLIP	
Epoch	10
Batch size	16
Learning rate	1e-4
Dropout	0.15
Weight Decay	1e-4
Embedding dimension	512
Projection dimension	256

	cMLIP result	
Group	Top1(%)	Top5(%)
Good Group1	100	100
Good Group2	100	100
Good Group3	100	100
Good Group4	100	100
Good Group5	100	100
Broke Group1	97	100
Broke Group2	100	100
Broke Group3	100	100
Broke Group4	100	100
Broke Group5	100	100
Shift Group1	76	100
Shift Group2	100	100
Shift Group3	100	100
Shift Group4	100	100
Shift Group5	100	100

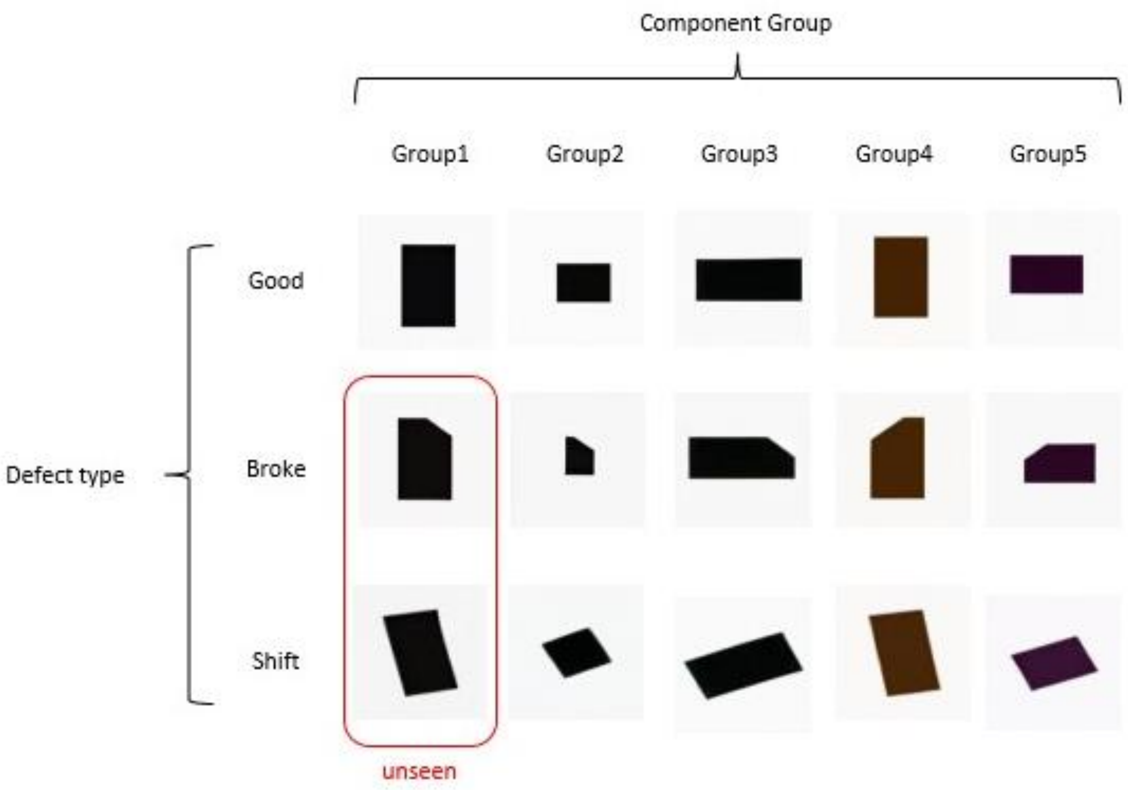
Toy Dataset

Compositional Conditional Diffusion Model			
Epoch	100	Depth	2
Batch size	64	Channel Multiplier	[1, 2, 2, 2]
Learning rate	5e-5	Head Channels	x
Optimizer	AdamW	Dropout	0.15
Diffusion steps	1000	Weight Decay	1e-4
Noise Schedule	linear	Embedding Dimension	512
Channel	128	Guidance strength(w)	1.8

Toy Dataset

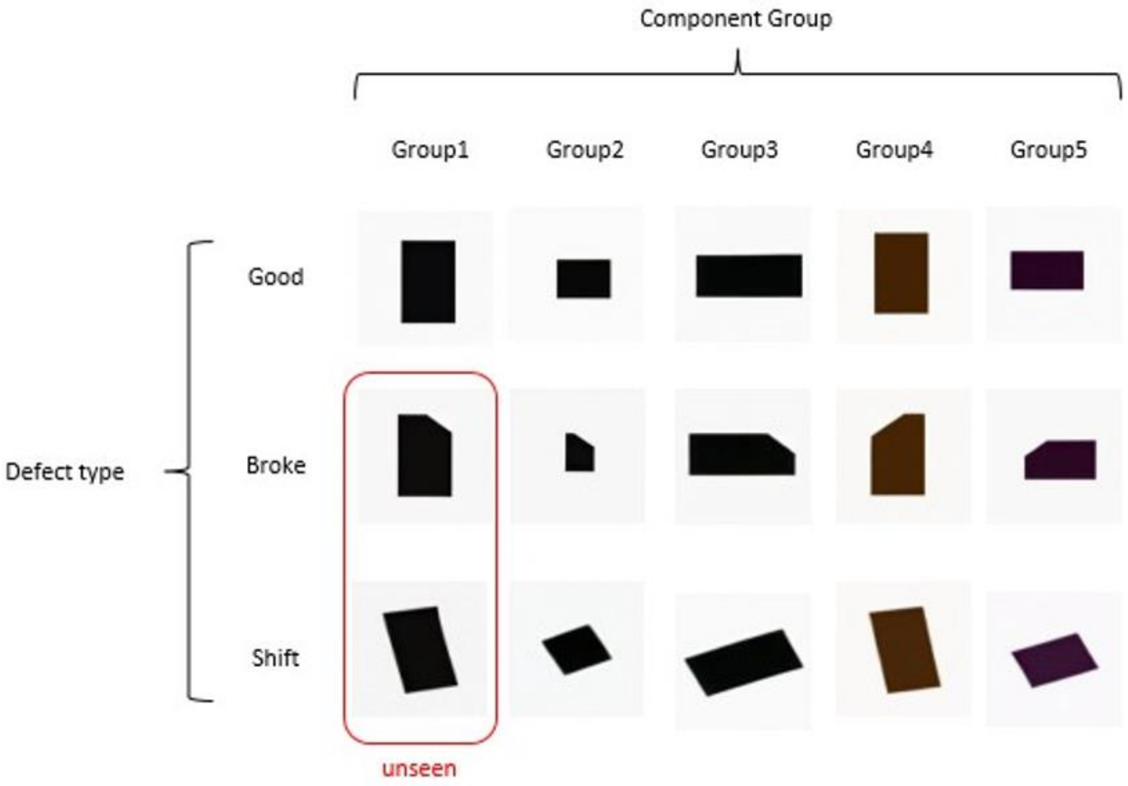


(a) U-Net_1(AddGN + learnable condition embedding)

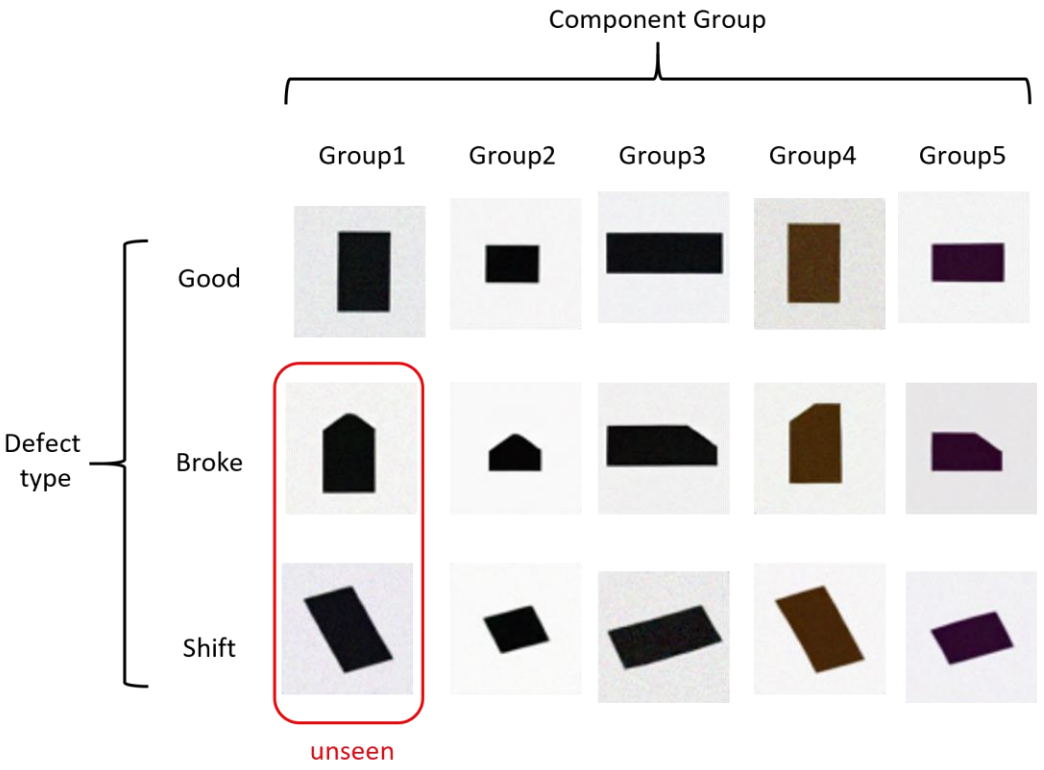


(b) U-Net_1(AddGN + fix condition embedding(random))

Toy Dataset

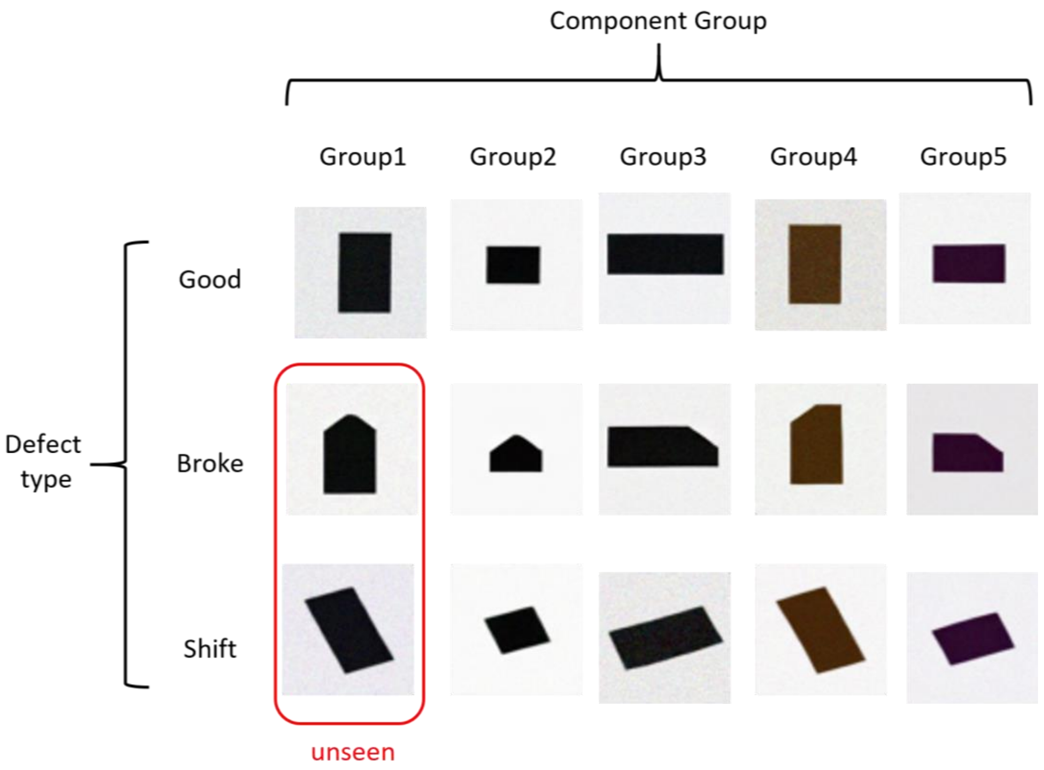


(b) U-Net_1(AddGN + fix condition embedding(random))

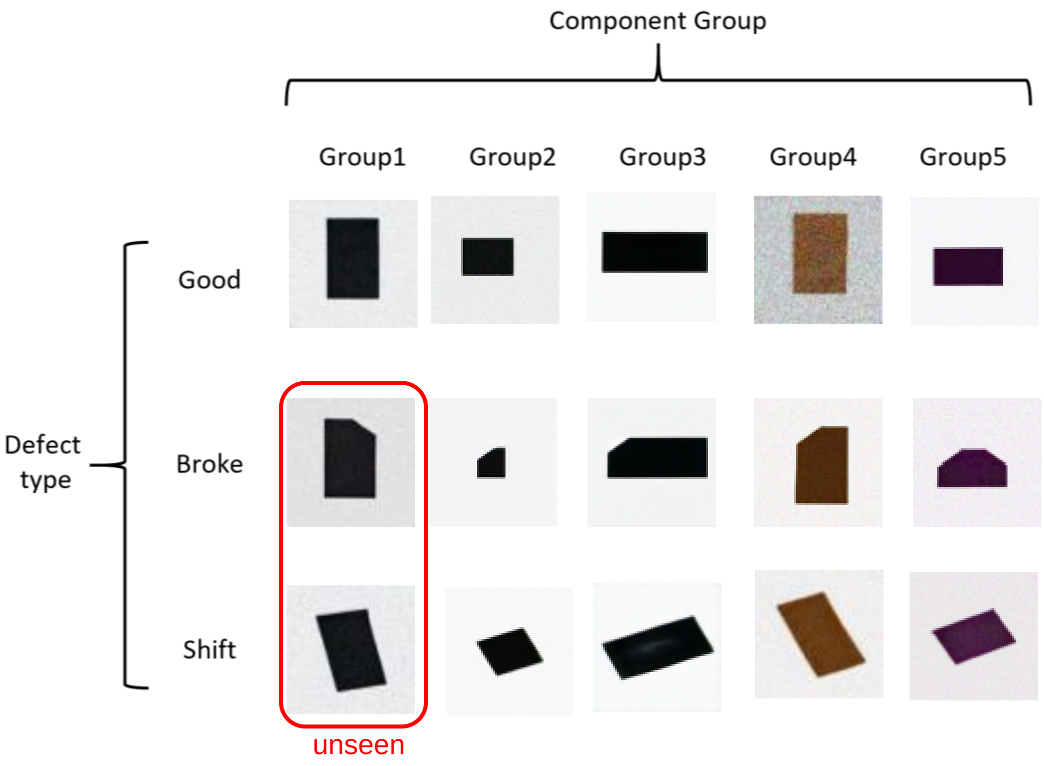


(c) U-Net_2(AddGN + fix condition embedding(cMLIP))

Toy Dataset



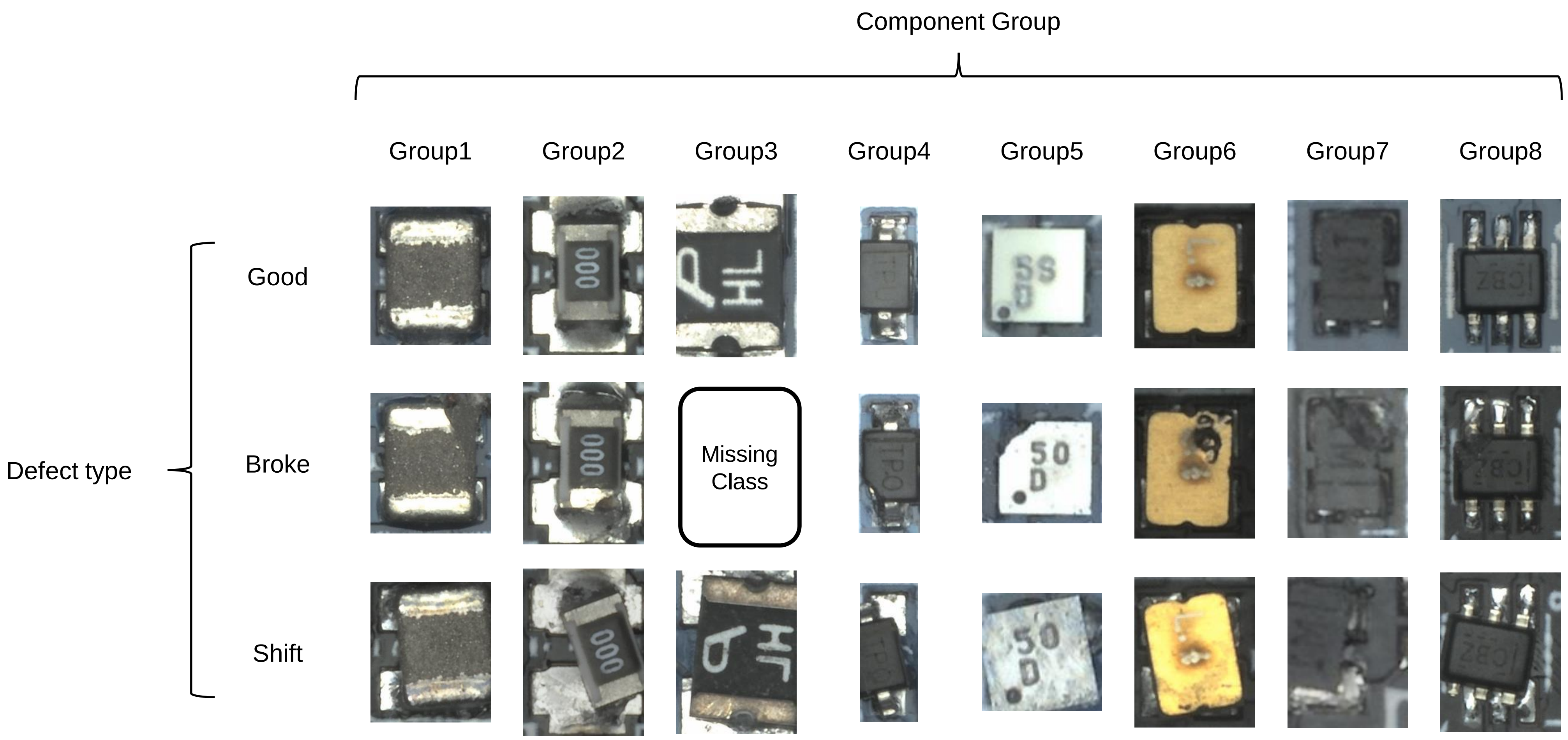
(c) U-Net_2(AddGN + fix condition embedding(cMLIP))



(d) U-Net_2(AdaGN + fix condition embedding(cMLIP))

PCB Dataset

PCB Dataset


Component Group
Group1
Group2
Group3
Group4
Group5
Group6
Group7
Group8
Defect type
Good
Broke
Shift
Missing Class

PCB Dataset

cMLIP	
Epoch	10
Batch size	16
Learning rate	1e-4
Dropout	0.15
Weight Decay	1e-4
Embedding dimension	512
Projection dimension	256

PCB Dataset

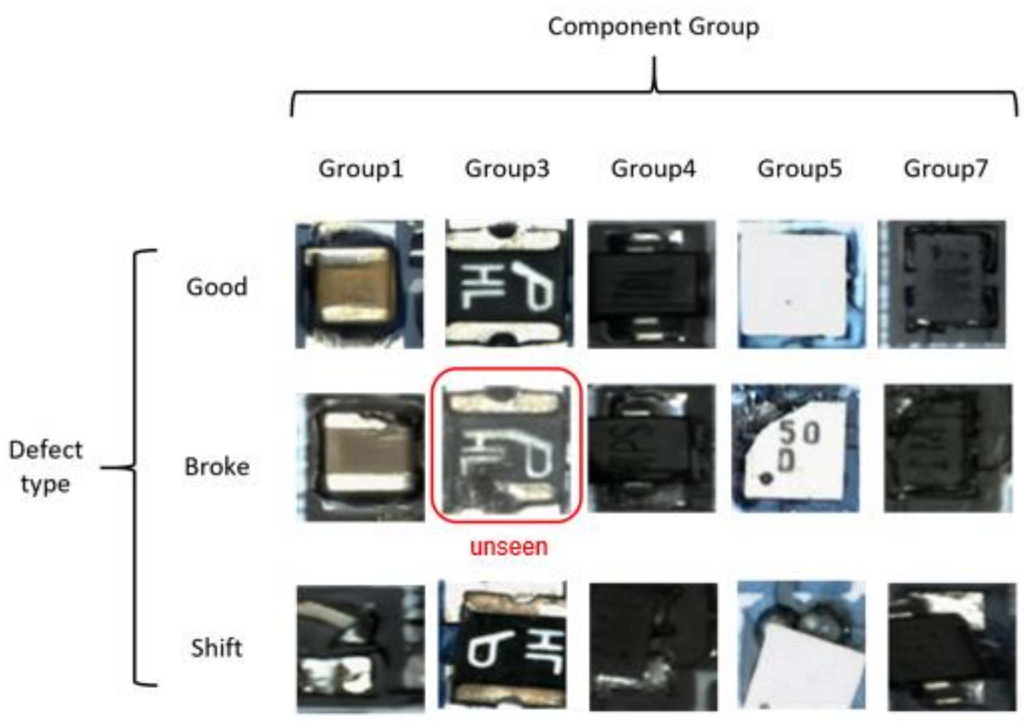
cMLIP	
Epoch	10
Batch size	16
Learning rate	1e-4
Dropout	0.15
Weight Decay	1e-4
Embedding dimension	512
Projection dimension	256

	cMLIP result	
Group	Top1(%)	Top5(%)
Good Group1	100	100
Good Group2	97	100
Good Group3	100	100
Good Group4	100	100
Good Group5	100	100
Broke Group1	96	100
Broke Group2	100	100
Broke Group3		
Broke Group4	100	100
Broke Group5	100	100
Shift Group1	83	100
Shift Group2	100	100
Shift Group3	100	100
Shift Group4	97	100
Shift Group5	99	100

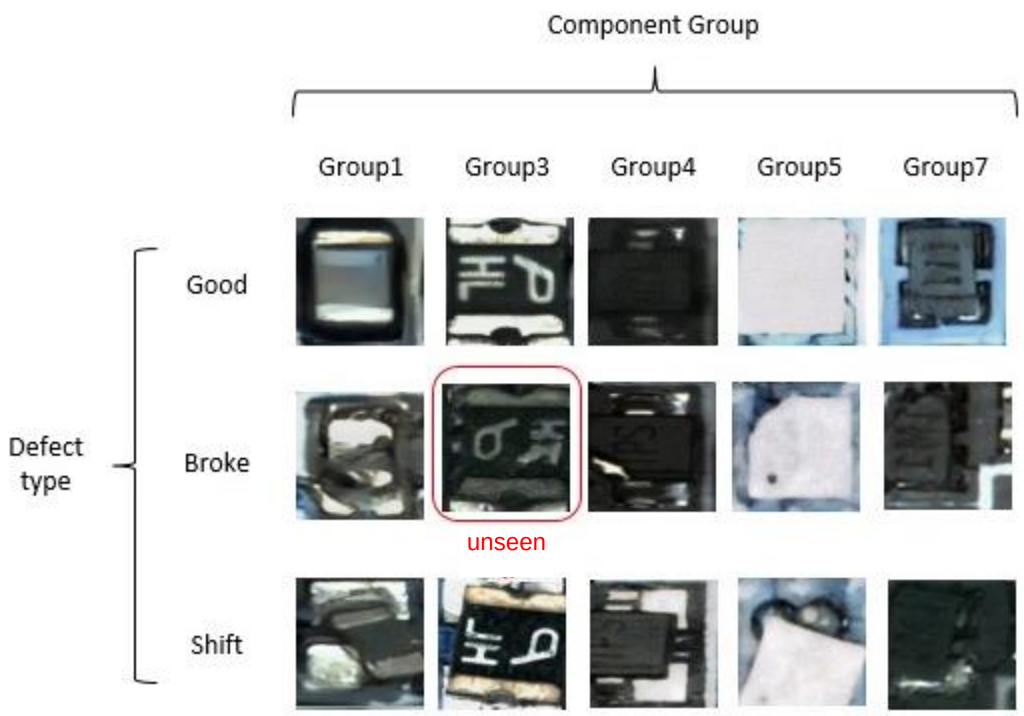
PCB Dataset

Compositional Conditional Diffusion Model			
Epoch	100	Depth	2
Batch size	64	Channel Multiplier	[1, 2, 2, 2]
Learning rate	4e-5	Head Channels	4
Optimizer	AdamW	Dropout	0.15
Diffusion steps	1000	Weight Decay	1e-4
Noise Schedule	linear	Embedding Dimension	512
Channel	128	Guidance strength(w)	5

PCB Dataset

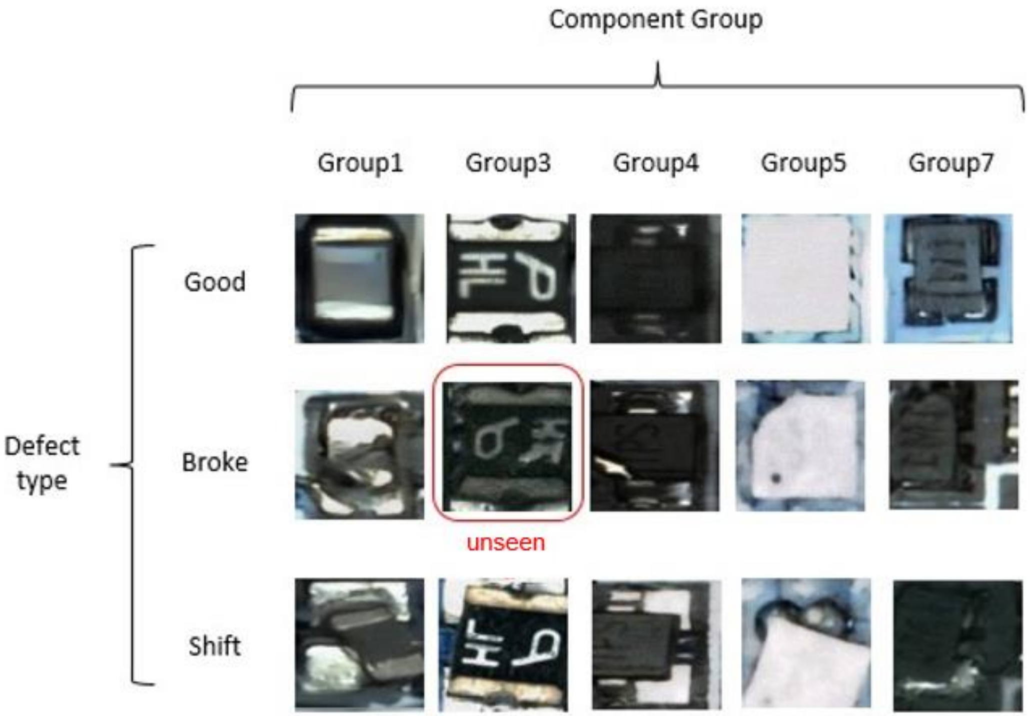


(a) U-Net_1(AddGN + learnable condition embedding)

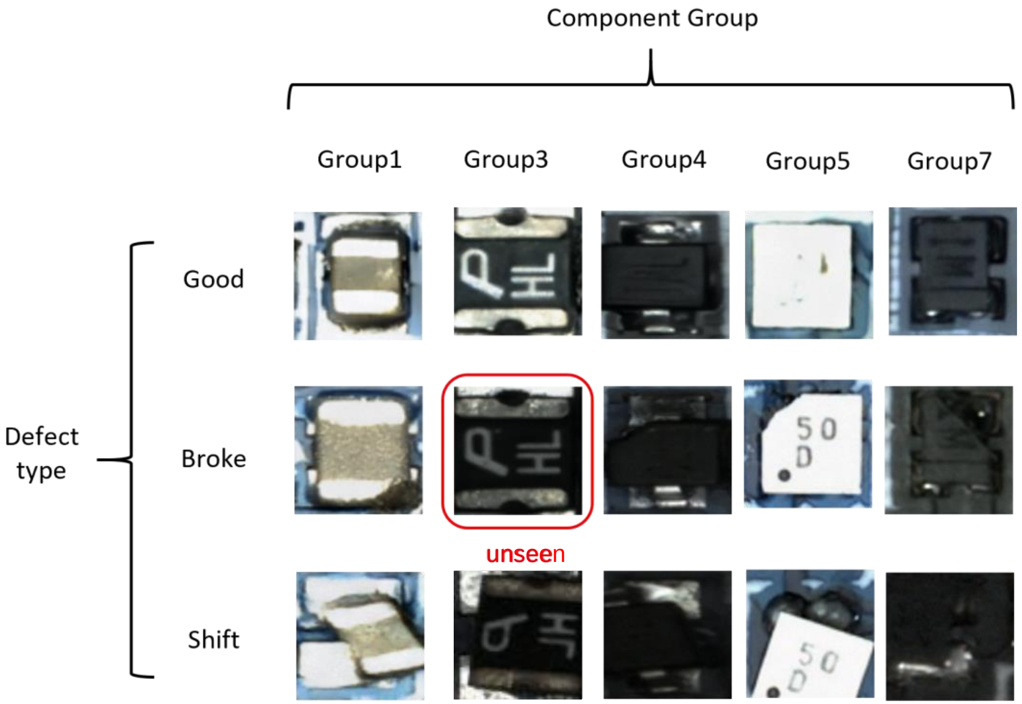


(b) U-Net_1(AddGN + fix condition embedding(random))

PCB Dataset

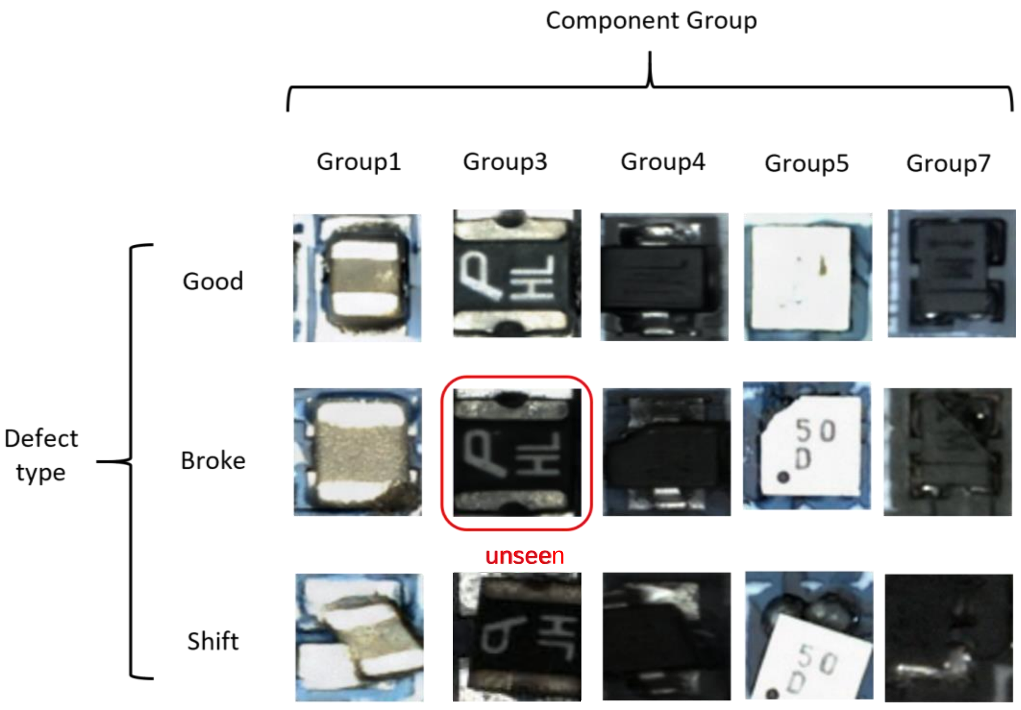


(b) U-Net_1(AddGN + fix condition embedding(random))

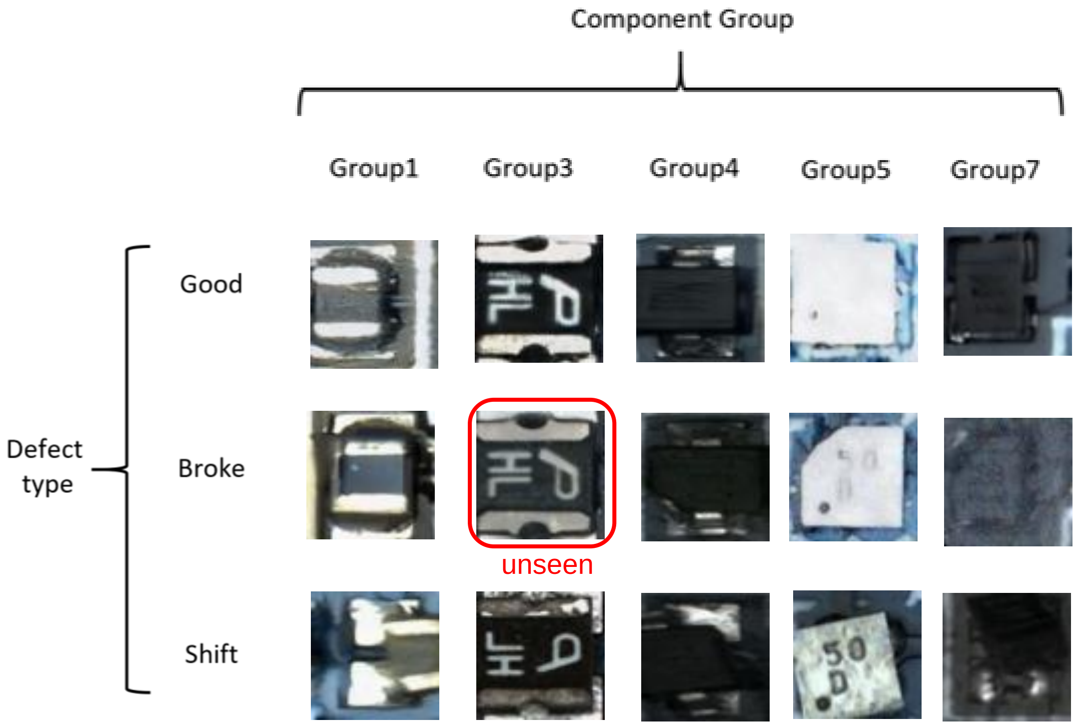


(c) U-Net_2(AddGN + fix condition embedding(cMLIP))

PCB Dataset



(c) U-Net_2(AddGN + fix condition embedding(cMLIP))

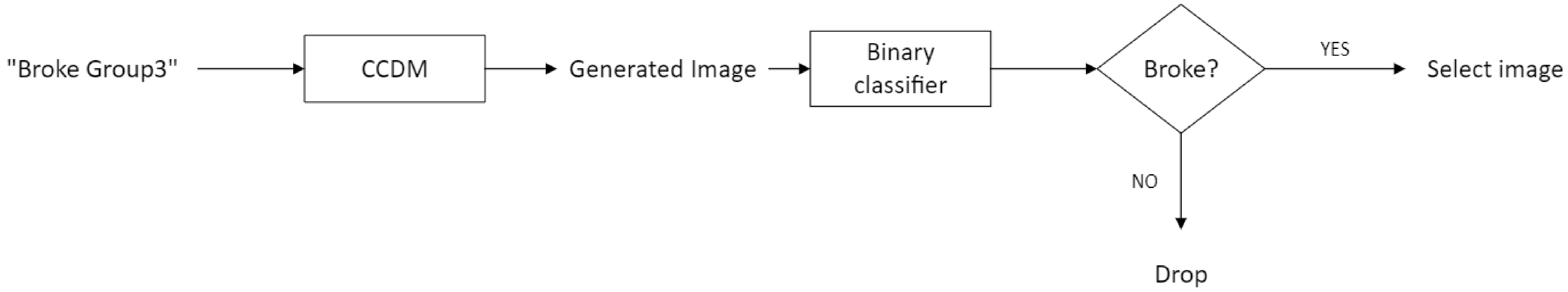


(d) U-Net_2(AdaGN + fix condition embedding(cMLIP))

PCB Dataset

Select Image

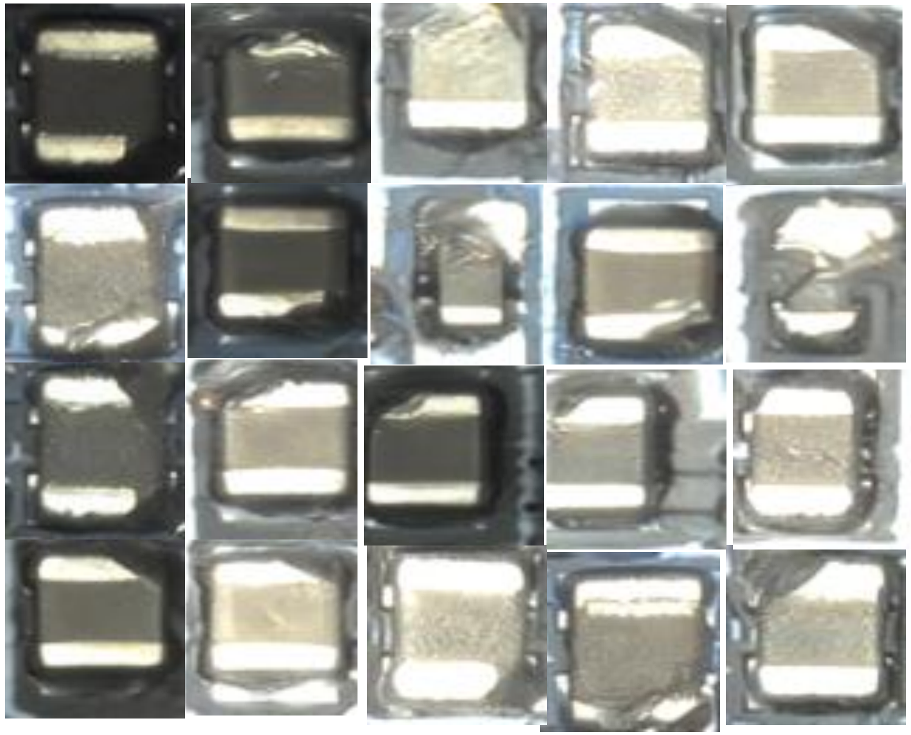
Select Image



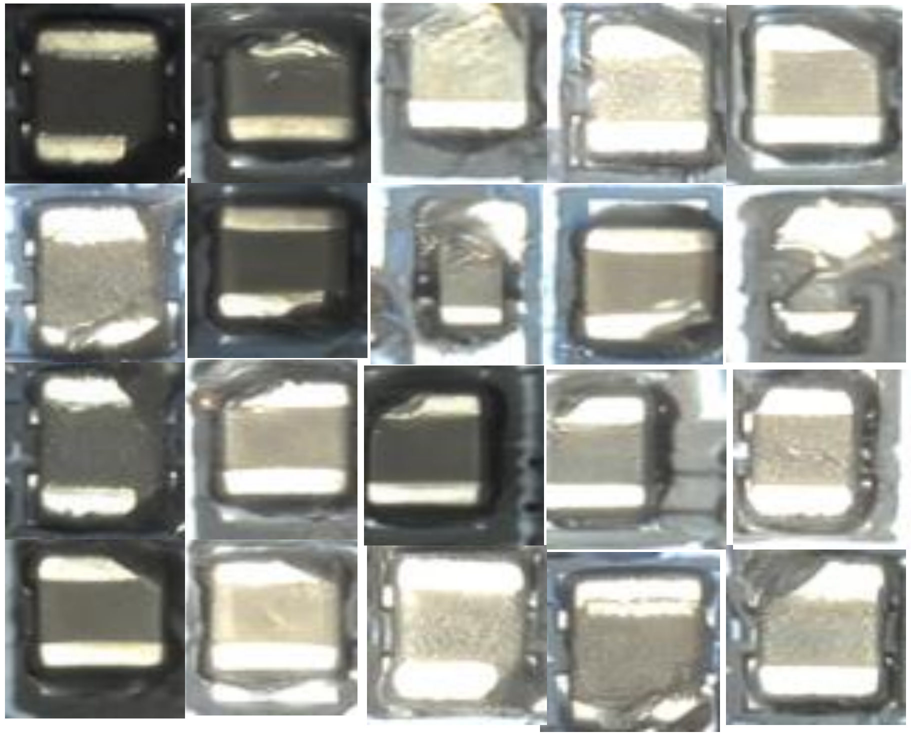
GOAL

Our goal is to utilize current image generation technology to create unseen defective components

Initially, without incorporating images of Broke Group1 generated by CCDM into the classification, the accuracy for correctly identifying real Broke Group1 was only about 22%. However, after adding the CCDM-generated images to the classifier for training, the accuracy improved to approximately 75%.



Finally, I added some raw images from Broke Group1 into the training data so that the dataset would not be entirely without Broke Group1, mimicking the typical structure of real data. Initially, the result was 97%. After incorporating photos generated by our CCDM as an augmented dataset, the accuracy improved to 98%.



Chapter5

Conclusion

Conclusion

- We propose an innovative concept focused on the generation of new compositional concepts in zero-shot learning.

- In Compositional Zero-Shot Learning, we emphasize learning from old compositional concepts and generalizing to new compositional concepts.

- Generate unseen defect components for extending the dataset used in PCB.

Thanks for your listening

Q & A