VARIATIONAL INFERENCE FOR CYCLIC LEARNING

Anonymous authors
Paper under double-blind review

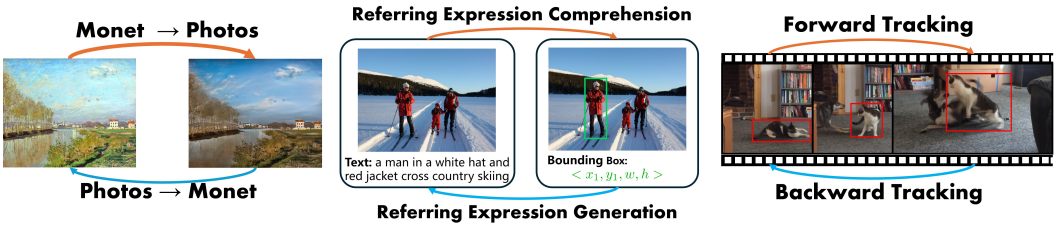
ABSTRACT

Cyclic learning, which involves training with pairs of inverse tasks and utilizes cycle-consistency in the design of loss functions, has emerged as a powerful paradigm for weakly-supervised learning. However, its potential remains under-explored due to the current methods' narrow focus on domain-specific implementations. In this work, we develop generalized solutions for both pairwise cycle-consistent tasks and self-cycle-consistent tasks. By formulating cross-domain mappings as conditional probability functions, we reformulate the cycle-consistency objective as an evidence lower bound optimization problem via variational inference. Based on this formulation, we further propose two training strategies for arbitrary cyclic learning tasks: single-step optimization and alternating optimization. Our framework demonstrates broad applicability across diverse tasks. In unpaired image translation, it not only provides a theoretical justification for CycleGAN but also leads to CycleGN—a competitive GAN-free alternative. For unsupervised tracking, CycleTrack and CycleTrack-EM achieve state-of-the-art performance on multiple benchmarks. This work establishes the theoretical foundations of cyclic learning and offers a general paradigm for future research.

1 INTRODUCTION

The need for labeled data is now one of the biggest obstacles in machine learning research, where supervised learning's reliance on manual labeling introduces both scalability issues and quality control challenges. To address this, researchers have turned to self-supervised training, the core idea of which is to generate supervisory signals from unlabeled data for training. **Self-consistency**-based self-supervised learning has already demonstrated strong capabilities in the field of representation learning(Zhang et al., 2016; He et al., 2022; Mikolov et al., 2013; Devlin et al., 2019; Chen & He, 2021). A series of studies have now shifted focus to cross-domain self-supervised learning constructed via **cyclic consistency**(Xu et al., 2023; Yuan et al., 2020; Dwibedi et al., 2019; Wang et al., 2024; Kulkarni et al., 2019).

This type of approach involves designing a pair of inverse tasks and constructing the training process by leveraging the property that data points should return to their origin after cyclic processing. This not only eliminates the reliance on manual annotations but also preserves task-specific semantic constraints. As shown in Fig. 1, this framework has been applied to various tasks(Zhu et al., 2017; Wang et al., 2024; 2019b; Dwibedi et al., 2019). A well-known example is CycleGAN(Zhu et al., 2017)(Fig. 1(a)), which jointly optimizes two tasks by combining cycle consistency loss and adversarial loss, leading to its widespread adoption in weakly supervised visual tasks(Almahairi et al., 2018; Yang et al., 2020; Kwon & Park, 2019). In contrast, a different approach is employed in the visual grounding (Referring Expression Comprehension) and image caption (Referring Expression Generation) loop(Fig. 1(b)), where both CyCO(Wang et al., 2024) and SC-Tune(Yue et al., 2024) adopt an alternating training strategy for the two tasks, showcasing the cross-modal adaptation capability. In CyCO, they first conduct cyclic training using only the cross-entropy loss with image captioning as the objective, followed by another training batch utilizes only the bounding-box losses(Rezatofighi et al., 2019; Girshick, 2015) for visual grounding. However, current approaches face two key limitations: First, task-specific designs hinder cross-domain generalization (e.g., the loss of CycleGAN cannot be directly applied to video alignment task). Second, many methods still rely on pseudo-labels (e.g., unsupervised visual tracking approaches(Zheng et al., 2021; Wang et al., 2019a; Shen et al., 2022) requiring initial trajectories from base trackers). To address this, we propose a probabilistic modeling approach to enable universal cyclic learning across all applicable tasks.



(a) Image-to-Image Translation (b) REC and REG (c) Visual Object Tracking

Figure 1: Tasks forming cyclic learning: (a) Image-to-image translation; (b) Referring expression comprehension & referring expression generation; (c) Visual object tracking.

Building upon the constraint of cycle consistency, this work establishes a unified probabilistic framework for both paired cyclic tasks (bidirectional mapping A→B and B→A) and self-cyclic tasks (A→B and B→A mapped by the same function). Methodologically, this framework is inspired by the Expectation-Maximization (EM) algorithm (Neal & Hinton, 1998), leveraging it as a foundational variational method for iterative training. By introducing a latent variable $\mathbf{z}$, it transforms the maximization of log-likelihood into the maximization of a evidence lower bound, and then approximates the optimum stepwise through the Expectation step (E-step) and Maximization step (M-step). This method has stood the test of time and remains highly influential across various fields to this day(Sun & Yang, 2020; Bao et al., 2024; Qu et al., 2019). Another canonical application is the Variational Autoencoder (VAE) (Kingma et al., 2013), which assumes that the latent variables corresponding to natural images follow the normal distribution. Through variational inference, it derives a reconstruction loss and a Kullback-Leibler divergence(Kullback & Leibler, 1951) loss, ultimately training a decoder capable of generating random images from standard Gaussian noise. In contrast, our framework aims to deliver theoretically rigorous and computationally efficient solutions for broad cyclic learning problems.

Specifically, we formalize cycle consistency by treating intermediate data points as latent variables, with cross-task transitions as learnable distributions. Within this framework, we propose: (**i**) a universal single-step loss derived via variational inference that enforces cycle consistency for end-to-end training, and (**ii**) an EM-based method that alternately updates model parameters in two tasks when KL divergence approximation is infeasible. To validate the universality and effectiveness of our method, we conduct experiments in two distinct tasks: for image translation, our approach not only reveals the working mechanism of CycleGAN(Zhu et al., 2017) but also achieves bidirectional style mappings without GANs through the EM method. In object tracking, our model effectively captures dynamic target appearance variations via self-cyclic constraints, significantly improving unsupervised tracking robustness. The proposed probabilistic framework provides a unified solution for diverse cyclic learning scenarios. Our main contributions are:

- We regard the intermediate points (non-starting/non-terminal points) in cyclic learning as latent variables, thereby establishing the first variational probabilistic framework that unifies both paired and self-cyclic tasks through variational inference.
- We derive two theoretically-grounded optimizers for general cyclic learning: (**i**) a single-step variational loss enabling stable and efficient training with explicit distributions, and (**ii**) a KL-free, EM-based algorithm compatible with complex distributions.
- In unpaired image translation, we theoretically explain the success of CycleGAN and propose a GAN-free, EM-based alternative. In visual tracking, we introduce CycleTrack (single-step) and CycleTrack-EM (EM-based), which achieve state-of-the-art unsupervised performance.

2 Variational Inference for Cyclic Learning

2.1 Methodology

Fundamentally, the generation problem involves learning a function f that maps data points from the input space to the output space, i.e., $f : \mathcal{X} \to \mathcal{Y}$. For example, in image captioning, $\mathcal{X}$ is a collection

of natural images where a data point $\mathbf{x}$ is a photo of a horse, and $\mathcal{Y}$ is the set of all grammatically correct sentences. The corresponding $\mathbf{y} \in \mathcal{Y}$ for $\mathbf{x}$ would be a natural language description of the horse. The goal of a generative model is to learn this mapping f, such that for an input $\mathbf{x}$ from the domain, the output $f(\mathbf{x})$ appears "real" and follows a specific distribution in the codomain $\mathcal{Y}$. Although the generative function itself is not an explicit probabilistic model, it implicitly encodes the dynamic process of probabilistic transition.

We now examine a special case from a theoretical perspective: when $\mathbf{y} = f(\mathbf{x})$ is invertible. In this scenario, f establishes a bijective mapping, and there exists a unique inverse function $\mathbf{x} = f^{-1}(\mathbf{y})$, which is a necessary condition to guarantee cycle consistency. When a specific observed value $\hat{\mathbf{y}}$ is given, it must have been produced by a unique $\hat{\mathbf{x}} = f^{-1}(\hat{\mathbf{y}})$. Probabilistically, this implies that under the condition $\mathcal{Y} = \{\mathbf{y}'\}$, the distribution of $\mathcal{X}$ is deterministic—all probability mass is concentrated at the single point $\mathbf{x}'$. Thus, the conditional probability density function $p(\mathbf{x}|\mathbf{y})$ becomes a Dirac function:

$$p(\mathbf{x}|\mathbf{y}) = \delta(\mathbf{x} - f^{-1}(\mathbf{y})). \tag{1}$$

Let $g(\cdot)$ denote $f^{-1}(\cdot)$, with ϕ and θ being the parameters to be learned for f and g respectively, the conditional probability can then be expressed as:

$$p_\theta(\mathbf{x}|\mathbf{y}) = \delta(\mathbf{x} - g_\theta(\mathbf{y})); \quad p_\phi(\mathbf{y}|\mathbf{x}) = \delta(\mathbf{y} - f_\phi(\mathbf{x})). \tag{2}$$

Based on the above transformation relationships, cyclical learning can be formulated probabilistically to optimize the mapping functions. First, considering the cycle starting from a data point $\mathbf{x}$ and returning to $\mathbf{x}$, we aim to maximize the log-likelihood, i.e., $\max \log p_\theta(\mathbf{x})$. By modeling the samples from domain $\mathcal{Y}$ as latent variables, we have:

$$\log p_\theta(\mathbf{x}) = \log \int q_\phi(\mathbf{y}|\mathbf{x}) p_\theta(\mathbf{x}) d\mathbf{y} \geq \mathbb{E}_{q_\phi(\mathbf{y}|\mathbf{x})} \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{y})}{q_\phi(\mathbf{y}|\mathbf{x})} \right] + D_{KL}(q_\phi(\mathbf{y}|\mathbf{x}) || p_{data}(\mathbf{y}|\mathbf{x})), \tag{3}$$

where D_{KL} denotes the Kullback-Leibler divergence, and $p_{data}(\mathbf{y}|\mathbf{x})$ represents the true conditional probability distribution of the output $\mathbf{y}$ given an input $\mathbf{x}$ in the real world. The first term corresponds to the so-called Evidence Lower Bound (ELBO), which admits the following decomposition:

$$\ell_{\theta,\phi}(\mathbf{x}) = \int q_\phi(\mathbf{y}|\mathbf{x}) \log p_\theta(\mathbf{x}|\mathbf{y}) d\mathbf{y} - D_{KL}(q_\phi(\mathbf{y}|\mathbf{x}) || p_{data}(\mathbf{y})), \tag{4}$$

where $p_{data}(\mathbf{y})$ denotes the true distribution. In Eq. 4, $\int q_\phi(\mathbf{y}|\mathbf{x}) \log p_\theta(\mathbf{x}|\mathbf{y}) d\mathbf{y}$ represents the reconstruction expectation, while $D_{KL}(q_\phi(\mathbf{y}|\mathbf{x})||p(\mathbf{y}))$ enforces distributional alignment between $q_\phi(\mathbf{y}|\mathbf{x})$ and the prior $p(\mathbf{y})$. One may simultaneously consider the symmetric case starting from a data point $\mathbf{y}$ and completing the cycle back to $\mathbf{y}$, for which the ELBO is given by:

$$\begin{aligned} \ell_{\theta,\phi}(\mathbf{x}, \mathbf{y}) \quad &= \int q_\phi(y|\mathbf{x}) \log p_\theta(\mathbf{x}|y) dy - D_{KL}(q_\phi(y|\mathbf{x}) || p_{data}(y)) \\ &+ \int q_\theta(x|\mathbf{y}) \log p_\phi(\mathbf{y}|x) dx - D_{KL}(q_\theta(x|\mathbf{y}) || p_{data}(x)). \end{aligned} \tag{5}$$

The gap between the maximum log-likelihood and its evidence lower bound is:

$$D_{KL}(q_\phi(\mathbf{y}|\mathbf{x}) || p_{data}(\mathbf{y}|\mathbf{x})) + D_{KL}(q_\theta(\mathbf{x}|\mathbf{y}) || p_{data}(\mathbf{x}|\mathbf{y})). \tag{6}$$

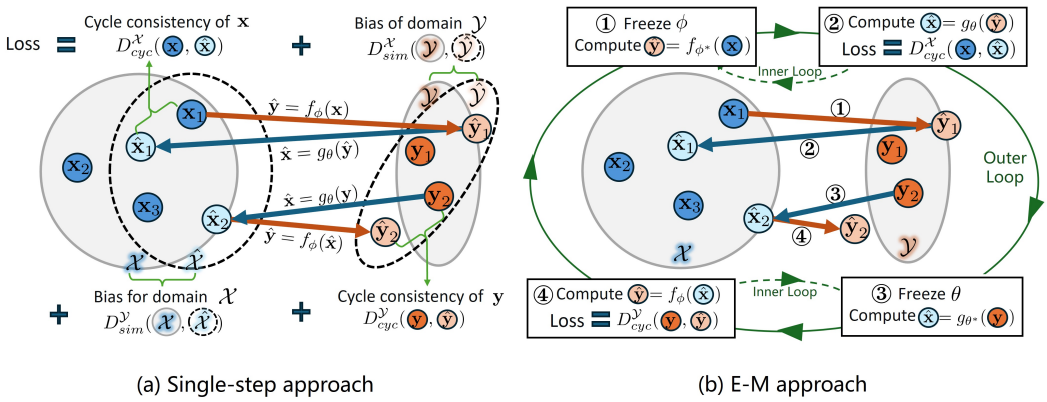
The maximization of $\ell_{\theta,\phi}(\mathbf{x}, \mathbf{y})$ inherently minimizes two KL divergence terms. Through this process, the approximation $q_\phi(\mathbf{y}|\mathbf{x})$ and $q_\theta(\mathbf{x}|\mathbf{y})$ progressively approach the true distributions $p_{data}(\mathbf{y}|\mathbf{x})$ and $p_{data}(\mathbf{x}|\mathbf{y})$, achieving exact alignment with the ultimate objective of cyclic learning. This variational inference process shares similarities with both VAE(Kingma et al., 2013) and EM(Neal & Hinton, 1998) algorithms. In fact, the two methods we will present next can be regarded as their direct counterparts: one being VAE-style and the other EM-style. The crucial difference in our framework is that instead of simply estimating $\mathbf{x}$'s distribution, we learn cross-domain mappings by analyzing $\mathbf{x}$'s distribution after cyclic transformation. Importantly, the latent variable $\mathbf{y}$ here is not freely designed but must strictly satisfy domain $\mathcal{Y}$'s constraints.

Returning to the perspective of mapping functions, for the first term in Eq. 4, we have:

$$\int q_\phi(\mathbf{y}|\mathbf{x}) \log p_\theta(\mathbf{x}|\mathbf{y}) d\mathbf{y} = \int \delta(\mathbf{y} - f_\phi(\mathbf{x})) \log \delta(\mathbf{x} - g_\theta(\mathbf{y})) d\mathbf{y} = \log \delta(\mathbf{x}, g_\theta(f_\phi(\mathbf{x}))), \tag{7}$$

Then for a chosen distance function $D_{cyc}(\mathbf{x}, \hat{\mathbf{x}})$ where $D_{cyc}(\mathbf{x}, \hat{\mathbf{x}}) = 0$ if $\mathbf{x} = \hat{\mathbf{x}}$, and $D_{cyc}(\mathbf{x}, \hat{\mathbf{x}}) > 0$ otherwise, the optimization of $\mathbb{E}_{q_\phi(\mathbf{y}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{y})]$ can be replaced with the optimization of D_{cyc}:

$$\arg\max_{\theta,\phi} \mathbb{E}_{q_\phi(\mathbf{y}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{y})] = \arg\max_{\theta,\phi} \log \delta(\mathbf{x}, g_\theta(f_\phi(\mathbf{x}))) = \arg\min_{\theta,\phi} D_{cyc}(\mathbf{x}, g_\theta(f_\phi(\mathbf{x}))). \tag{8}$$



(a) Single-step approach (b) E-M approach

Figure 2: General solution for cyclic learning. (a) Single-step direct optimization. (b) Dual EM-iteration optimization.

Here we introduce D_{cyc} to approximate the expectation in order to fulfill task-specific requirements for similarity measurement. The core idea is to minimize the discrepancy between $g_\theta\left(f_\phi(\mathbf{x})\right)$ and $\mathbf{x}$, as exemplified by the Intersection over Union function in bounding box regression tasks.

For the second term in Eq. 4, we have:

$$D_{KL}(q_\phi(\mathbf{y}|\mathbf{x})||p_{data}(\mathbf{y})) = \int \delta(\mathbf{y} - f_\phi(\mathbf{x})) \cdot log\frac{\delta(\mathbf{y} - f_\phi(\mathbf{x}))}{p_{data}(\mathbf{y})}d\mathbf{y} = \log\delta(0) - \log p_{data}(f_\phi(\mathbf{x})). \tag{9}$$

where $\log\delta(0)$ is a divergent constant, while $\log p_{data}(f_\phi(\mathbf{x}))$ measures the probability that the generated output $f_\phi(\mathbf{x})$ conforms to the distribution $p_{data}(\mathbf{y})$, representing the optimization objective of the KL divergence constraint. Similarly, if $p(\mathbf{y})$ is undefined, the distance function $D_{sim}(\hat{\mathbf{y}}, \mathcal{Y})$ can be introduced to quantify the difference between the generated $\hat{\mathbf{y}}$ and the target set $\mathcal{Y}$, serving as a proxy for $\log p_{data}(f_\phi(\mathbf{x}))$:

$$\arg\max_\phi \; -D_{KL}(q_\phi(\mathbf{y}|\mathbf{x})||p_{data}(\mathbf{y})) = \arg\max_\phi \; \log p_{data}(f_\phi(\mathbf{x})) = \arg\min_\phi D_{sim}(f_\phi(\mathbf{x}), \mathcal{Y}). \tag{10}$$

A classic choice for D_{sim} is the Wasserstein distance (Vaserstein, 1969). By combining Eq. 4 with Eq. 8 and Eq. 10, the optimization process of the ELBO can be expressed as

$$\arg\max_{\theta,\phi} \ell_{\theta,\phi}(\mathbf{x}) \approx \arg\min_{\theta,\phi} \left(D_{cyc}(\mathbf{x}, g_\theta(f_\phi(\mathbf{x}))) + D_{sim}(f_\phi(\mathbf{x}), \mathcal{Y})\right). \tag{11}$$

Note that $\ell_{\theta,\phi}(\mathbf{x})$ requires joint optimization of θ and ϕ, which may not achieve exact equality in Eq. 11 due to differing gradient behaviors across distance metrics. Nevertheless, since both D_{cyc} and D_{sim} are proxy methods, maximizing $D_{cyc} + D_{sim}$ to approximate $\max \ell_{\theta,\phi}(\mathbf{x})$ is not detrimental to training. The core design principle requires the loss function to incorporate:

- The similarity between $\hat{\mathbf{x}}$ and $\mathbf{x}$ (measured by D_{cyc})
- The degree to which $\hat{\mathbf{y}}$ belongs to $\mathcal{Y}$ (measured by D_{sim})

For cyclic tasks, models may converge to local optima of either $\min D_{cyc}$ or $\min D_{sim}$. This necessitates task-specific balancing between these two approximating terms.

By integrating Eq. 5 with Eq. 11, we arrive at the two-way cycle-consistent loss:

$$\mathcal{L}(\mathbf{x}, \mathbf{y}) = D^{\mathcal{X}}_{cyc}(\mathbf{x}, g_\theta(f_\phi(\mathbf{x}))) + D^{\mathcal{X}}_{sim}(f_\phi(\mathbf{x}), \mathcal{Y}) + D^{\mathcal{Y}}_{cyc}(\mathbf{y}, f_\phi(g_\theta(\mathbf{y}))) + D^{\mathcal{Y}}_{sim}(g_\theta(\mathbf{y}), \mathcal{X}), \tag{12}$$

which corresponds to the direct optimization scheme shown in Fig. 2(a).

Building on the favorable properties of latent-variable-like designs, we also propose an EM-based method that alternately maximizes the log-likelihoods of $p_\theta(\mathbf{x})$ and $p_\phi(\mathbf{y})$ which is illustrated in Fig. 2(b). This cyclic learning process consists of two alternating stages, termed E_θ-M_θ and E_ϕ-M_ϕ, which are designed to promote global convergence. Specifically, the E_θ-M_θ stage optimizes the parameters θ, while the E_ϕ-M_ϕ stage optimizes ϕ.

In the E_θ-M_θ stage, we assume the true distribution $p_{\text{data}}(\mathbf{y}|\mathbf{x})$ is equal to $p_{\phi^*}(\mathbf{y}|\mathbf{x})$. Under this assumption, the E_θ step enforces $D_{\text{KL}}(q_\phi(\mathbf{y}|\mathbf{x})\|p_{\phi^*}(\mathbf{y}|\mathbf{x})) = 0$, which implies $q_\phi(\mathbf{y}|\mathbf{x}) = p_{\phi^*}(\mathbf{y}|\mathbf{x})$. Subsequently, the M_θ step updates θ by maximizing $\mathbb{E}_{p_{\phi^*}}(\mathbf{y}|\mathbf{x})[\log p_\theta(\mathbf{x}|\mathbf{y})]$, which drives θ to converge towards a lower bound determined by the approximation quality of $p_{\phi^*}(\mathbf{y}|\mathbf{x})$. In the E_ϕ-M_ϕ stage, we assume that θ^* is sufficiently accurate, such that $p_{\theta^*}(\mathbf{x}|\mathbf{y})$ effectively represents the true conditional distribution $p_{\text{data}}(\mathbf{x}|\mathbf{y})$. Analogously, the E_ϕ step enforces $D_{\text{KL}}(q_\theta(\mathbf{x}|\mathbf{y})\|p_{\theta^*}(\mathbf{x}|\mathbf{y})) = 0$, and the M_ϕ step updates ϕ by maximizing the expectation $\mathbb{E}_{p_{\theta^*}(\mathbf{x}|\mathbf{y})}[\log p_\phi(\mathbf{y}|\mathbf{x})]$. Algo. 1 details the procedure described above, but with the tone of generative models.

Algorithm 1 An EM approach for cycle-consistent tasks.

Input: Dataset $\mathcal{X} = \{\mathbf{x}^i\}_{i=1}^N$, $\mathcal{Y} = \{\mathbf{y}^i\}_{i=1}^M$

while not converge **do**
  while insufficient loss decrease **do**
    Sample batch of datapoints $\mathcal{X}' = \{\mathbf{x}\}$ from $\mathcal{X}$
    Using ϕ to get $\mathcal{T}' = \{(\mathbf{x}, \hat{\mathbf{y}})|\hat{\mathbf{y}} = f_\phi(\mathbf{x})\}$ \\ E_θ
    Update θ via $\mathcal{L}(\theta) = \sum_{\mathcal{T}'} D_{cyc}^{\mathcal{X}}(\mathbf{x}, g_\theta(\hat{\mathbf{y}}))$ \\ M_θ
  end while
  while insufficient loss decrease **do**
    Sample batch of datapoints $\mathcal{Y}' = \{\mathbf{y}\}$ from $\mathcal{Y}$
    Using θ to get $\mathcal{T}' = \{(\hat{\mathbf{x}}, \mathbf{y})|\hat{\mathbf{x}} = g_\theta(\mathbf{y})\}$ \\ E_ϕ
    Update ϕ via $\mathcal{L}(\phi) = \sum_{\mathcal{T}'} D_{cyc}^{\mathcal{Y}}(\mathbf{y}, f_\phi(\hat{\mathbf{x}}))$ \\ M_ϕ
  end while
end while

Output: Generative models $g_\theta(\cdot)$ and $f_\phi(\cdot)$.

In the standard EM algorithm, the E-step fixes parameters θ to find the latent distribution $q(\mathbf{z})$ that tightens the ELBO, typically by computing the posterior $p(\mathbf{z}|\mathbf{x}, \theta)$. The M-step then fixes $q(\mathbf{z})$ and updates θ to maximize the expected complete-data log-likelihood. This process can be viewed as coordinate ascent, alternately optimizing θ and $q(\mathbf{z})$. Our method employs a different variational inference workflow since we focus on learning mappings $p(\mathbf{x}|\mathbf{y})$ and $p(\mathbf{y}|\mathbf{x})$ rather than data distributions. When optimizing $p_\theta(\mathbf{x}|\mathbf{y})$, the E-step fixes θ (via E_ϕ) and finds the optimal $q_\phi(\mathbf{y}|\mathbf{x})$ (via M_ϕ), while the M-step maximizes the expected log-likelihood under this $q_\phi(\mathbf{y}|\mathbf{x})$ (via E_θ and M_θ). The same logic applies to $p_\phi(\mathbf{y}|\mathbf{x})$. Parameters θ and ϕ are alternately updated while keeping the other fixed. For clarity, we define fixing network parameters and sampling data as the E-step, and maximizing distribution functions as the M-step, labeled as E_ϕ-M_ϕ and E_θ-M_θ based on the optimized parameters. Essentially, E_ϕ-M_ϕ serves as the E-step for $p(\mathbf{x}|\mathbf{y})$ while E_θ-M_θ acts as its M-step. Thus, our method also implements coordinate ascent, alternately optimizing model parameters and parameterized distribution functions.

This approach eliminates the need to define D_{sim} or estimate $p_{data}(\cdot)$ as in Eq. 12, thus avoiding both training instability caused by metric inaccuracies or ill-defined data distributions. The alternating nature of EM optimization prevents direct distribution control through D_{KL}. Its convergence guarantee for $\hat{\mathbf{y}} = f_\phi(\mathbf{x}) \in \mathcal{Y}$ stems from the M-step's enforcement of $\hat{\mathbf{y}} = f_\phi(g_\theta(\mathbf{y}))$ to approximate $\mathbf{y}$, which is similar to SimSiam(Chen & He, 2021). But in practice, the EM method does carry a risk of converging to local optima due to the lack of explicit constraints on latent variables.

2.2 Application on Unpaired Image Translation

We use CycleGAN(Zhu et al., 2017) as an example and conduct experiments on the unpaired image-to-image translation task. For two distinct image domains $\mathcal{X}$ and $\mathcal{Y}$, the objective of CycleGAN is to find a pair of mapping functions: $f_\phi : \mathcal{X} \to \mathcal{Y}$ and $g_\theta : \mathcal{Y} \to \mathcal{X}$. The method employs a single-step optimization strategy for network training, with the proposed loss function as follows:

$$\mathcal{L}(f_\phi, g_\theta, D_\mathcal{X}, D_\mathcal{Y}) = \mathcal{L}_{\text{GAN}}(f_\phi, D_\mathcal{Y}, \mathcal{X}, \mathcal{Y}) + \mathcal{L}_{\text{GAN}}(g_\theta, D_\mathcal{X}, \mathcal{Y}, \mathcal{X}) + \mathcal{L}_{\text{cyc}}(f_\phi, g_\theta), \tag{13}$$

where $\mathcal{L}_{\text{GAN}}$ denotes the adversarial loss and $\mathcal{L}_{\text{cyc}}$ represents the cycle consistency loss. $D_\mathcal{X}$ and $D_\mathcal{Y}$ are the discriminators for domains $\mathcal{X}$ and $\mathcal{Y}$ respectively, which engage in adversarial training with the generators G and f_ϕ. $\mathcal{L}_{\text{cyc}}$ consists of both forward and backward cycle consistency loss:

$$\mathcal{L}_{\text{cyc}}(f_\phi, g_\theta) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\|g_\theta(f_\phi(\mathbf{x})) - \mathbf{x}\|_1] + \mathbb{E}_{\mathbf{y}\sim p_{data}(\mathbf{y})}[\|f_\phi(g_\theta(\mathbf{y})) - \mathbf{y}\|_1]. \tag{14}$$

Under our framework, the corresponding components of the loss function can be mapped in Eq.12 shown in Tab. 1. It can be seen that $\mathcal{L}_{\text{GAN}}$ enforces similarity between the generated distribution and the target distribution, while $\mathcal{L}_{\text{cyc}}$ enforces cycle consistency. The discriminator in a GAN is related to the Jensen-Shannon (JS) divergence(Lin, 2002) between the generated and real data distributions(Goodfellow et al., 2014). As the JS divergence itself is a symmetric reformulation of the

Table 1: The correspondence between components in Eq.12 and those in CycleGAN.

Components in Eq. 12	Components in CycleGAN
$D^{\mathcal{X}}_{cyc}(\mathbf{x}, g_\theta(f_\phi(\mathbf{x})))$	$\mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\|\|g_\theta(f_\phi(\mathbf{x})) - \mathbf{x}\|\|_1]$
$D^{\mathcal{X}}_{sim}(f_\phi(\mathbf{x}), \mathcal{Y})$	$\mathcal{L}_{\text{GAN}}(f_\phi, D_\mathcal{Y}, \mathcal{X}, \mathcal{Y})$
$D^{\mathcal{Y}}_{cyc}(\mathbf{y}, f_\phi(g_\theta(\mathbf{y})))$	$\mathbb{E}_{\mathbf{y}\sim p_{data}(\mathbf{y})}[\|\|f_\phi(g_\theta(\mathbf{y})) - \mathbf{y}\|\|_1]$
$D^{\mathcal{Y}}_{sim}(g_\theta(\mathbf{y}), \mathcal{X})$	$\mathcal{L}_{\text{GAN}}(g_\theta, D_\mathcal{X}, \mathcal{Y}, \mathcal{X})$

Table 2: The correspondence between steps in Algo. 1 and those in CycleGN.

Steps in Algo. 1.	Steps in CycleGN
E_θ	$\hat{\mathbf{y}} = f_\phi(\mathbf{x})$
M_θ	Update θ via $\mathcal{L}_{\text{cyc}}(g_\theta) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\|\|g_\theta(\hat{\mathbf{y}}) - \mathbf{x}\|\|_1]$
E_ϕ	$\hat{\mathbf{x}} = g_\theta(\mathbf{y})$
M_ϕ	Update ϕ via $\mathcal{L}_{\text{cyc}}(f_\phi) = \mathbb{E}_{\mathbf{y}\sim p_{data}(\mathbf{y})}[\|\|f_\phi(\hat{\mathbf{x}}) - \mathbf{y}\|\|_1]$

KL divergence, the adversarial loss function effectively serves as an indirect method for approximating D_{KL}. Integrating these four components reveals that Eq. 13 is essentially an application of Eq. 12, which explains why CycleGAN works effectively.

Based on Algo. 1, we propose a cyclic learning approach that alternately optimizes tasks in both directions. The detailed procedure is summarized in Tab. 2, with iterations continuing until convergence is achieved. Unlike CycleGAN, our method removes adversarial discriminators entirely and is thus named CycleGN.

2.3 Experiments

Table 3: FCN-scores of labels→photo for different methods on Cityscapes.

Loss	GAN	Per-pixel acc.	Per-class acc.	Class IOU
CoGAN	✓	<u>0.40</u>	0.10	0.06
BiGAN/ALI	✓	0.19	0.06	0.02
SimGAN	✓	0.20	0.10	0.04
Feat. loss + GAN	✓	0.06	0.04	0.01
CycleGAN	✓	**0.52**	**0.17**	**0.11**
CycleGN (ours)	×	**0.52**	<u>0.14</u>	<u>0.10</u>

Table 4: Classification performance of photo→labels on Cityscapes.

Loss	GAN	Per-pixel acc.	Per-class acc.	Class IOU
CoGAN	✓	0.45	0.11	0.08
BiGAN/ALI	✓	0.41	0.13	0.07
SimGAN	✓	0.47	0.11	0.07
Feat. loss + GAN	✓	0.50	0.10	0.06
CycleGAN	✓	**0.58**	**0.22**	**0.16**
CycleGN (ours)	×	<u>0.51</u>	<u>0.16</u>	<u>0.10</u>

CycleGN and CycleGAN(Zhu et al., 2017) use the same generator network from pix2pix(Isola et al., 2017). The unpaired training and test sets are from Cityscapes(Cordts et al., 2016). CycleGN switches between training E_θ-M_θ and E_ϕ-M_ϕ every 200 data samples, with a total of 100 training epochs. All other training settings remain consistent with CycleGAN. We compare approaches employing different loss functions for cyclic learning, with experimental results reported by CycleGAN, including: CoGAN(Liu & Tuzel, 2016), BiGAN/ALI(Dumoulin et al., 2016; Donahue et al., 2016), SimGAN(Shrivastava et al., 2017), Feature loss + GAN(Shrivastava et al., 2017; Zhu et al., 2017). We conduct experiments for labels-to-photo and photo-to-labels translation on Cityscapes. Tab. 3 and 4 compare CycleGN with CycleGAN and other loss configurations. The successes achieved by CycleGAN on this pair of cyclic tasks demonstrate the feasibility of the single-step optimization paradigm. Moreover, our proposed application of CycleGN based on the EM method on this task achieves better accuracy than other loss functions, with only a minor gap compared to CycleGAN. Notably, CycleGN does not even employ an adversarial structure—competitive generation results were still obtained by pushing the outputs of the generative network closer to instances within the target domain.

3 Extending to Self-Cyclic Learning

3.1 Methodology

We consider a special case with self-cycle-consistent single-task learning, where for any $\mathcal{X}, \mathcal{Y} \in \Omega$, and any $\mathbf{x} \in \mathcal{X}$, $\mathbf{y} \in \mathcal{Y}$, the symmetry $p(\mathbf{x}|\mathbf{y}) = p(\mathbf{y}|\mathbf{x})$ holds (i.e., $f_\phi = g_\theta$). When $f_\phi = g_\theta$, a trivial solution $g_\theta(g_\theta(\mathbf{x})) = \mathbf{x}$ would be $g_\theta(\mathbf{x}) = \mathbf{x}$. However, since $\hat{\mathbf{y}} = g_\theta(\mathbf{x})$ must belong to $\mathcal{Y}$ not $\mathcal{X}$ in cyclic learning, $g_\theta(\mathbf{x})$ cannot directly equal $\mathbf{x}$. Thus, we reformulate the optimization objective of cyclical learning as $g_\theta(g_\theta(\mathbf{x}, \mathcal{X}, \mathcal{Y}), \mathcal{Y}, \mathcal{X}) = \mathbf{x}$, which corresponds to optimizing the conditional

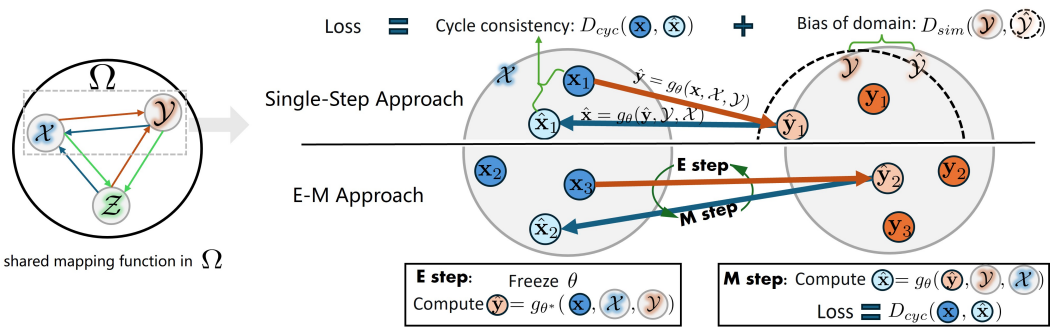


Figure 3: General solution for self-cycle-consistent learning.

probability $\log p_\theta(\mathbf{x}|\mathcal{Y}, \mathcal{X})$, yielding the evidence lower bound:

$$\ell_\theta(\mathbf{x}|\mathcal{Y}, \mathcal{X}) = \mathbb{E}_{q_\theta(\mathbf{y}|\mathbf{x},\mathcal{X},\mathcal{Y})}\left[\log p_\theta(\mathbf{x}|\mathbf{y}, \mathcal{Y}, \mathcal{X})\right] - D_{KL}(q_\theta(\mathbf{y}|\mathbf{x}, \mathcal{X}, \mathcal{Y})||p_{data}(\mathbf{y})). \tag{15}$$

For self-cycle tasks, either the backward or forward components in the asymmetric task loss in Eq. 12 can be removed, leading to the follow loss function:

$$\mathcal{L}(\mathrm{x}) = D_{cyc}(\mathbf{x}, g_\theta(g_\theta(\mathbf{x}, \mathcal{X}, \mathcal{Y}), \mathcal{Y}, \mathcal{X})) + D_{sim}(g_\theta(\mathbf{x}, \mathcal{X}, \mathcal{Y}), \mathcal{Y}). \tag{16}$$

This loss function guides the model $\hat{\mathbf{y}} = g_\theta(\cdot, \cdot, \mathcal{Y})$ toward $\hat{\mathbf{y}} \in \mathcal{Y}$ through D_{sim}, preventing convergence to the local optimum $g_\theta(\mathbf{x}, \mathcal{X}, \cdot) = \mathbf{x}$.

Algorithm 2 An EM approach for self-cycle-consistent tasks.

Input: Dataset $\Omega = \{\mathcal{X}^i\}_{i=1}^{\mathbf{N}}$ with $\mathcal{X}^i = \{\mathbf{x}^{ij}\}_{j=1}^{N^i}$

1: **while** not converge **do**
2: Random chosen sub-domains $\mathcal{X}, \mathcal{Y} \subseteq \Omega$
3: Sample batch of datapoints $\mathcal{X}' = \{\mathbf{x}\}$ from $\mathcal{X}$
4: ⊡ **E-step: Stop Gradient**
5: Using θ to get $\mathcal{T}' = \{(\mathbf{x}, \hat{\mathbf{y}})|\hat{\mathbf{y}} = g_\theta(\mathbf{x}, \mathcal{X}, \mathcal{Y})\}$
6: ⊡ **M-step: Learning Procedure**
7: Update θ via $\mathcal{L}(\theta) = \sum_{\mathcal{T}'} D_{cyc}\left(\mathbf{x}, g_\theta(\hat{\mathbf{y}}, \mathcal{Y}, \mathcal{X})\right)$
8: **end while**

Output: Generative model $g_\theta(\cdot)$.

The EM algorithm remains applicable to self-cycle tasks. As outlined in Algo. 2, the E-step minimizes $D_{KL}(q_\theta(\mathbf{y}|\mathbf{x}, \mathcal{X}, \mathcal{Y})||p_{\theta^*}(\mathbf{y}|\mathbf{x}, \mathcal{X}, \mathcal{Y}))$, while the M-step performs posterior probability maximization via $\arg\max \mathbb{E}_{p_{\theta^*}(\mathbf{y}|\mathbf{x},\mathcal{X},\mathcal{Y})}\left[\log p_\theta(\mathbf{x}|\mathbf{y}, \mathcal{Y}, \mathcal{X})\right]$. The KL divergence term in Eq. 15 is omitted in M-step as it reduces to a constant in this case. Note that $f_\phi = g_\theta$ enables joint optimization of bidirectional tasks in one EM process, contrasting with the E_θ-M_θ / E_ϕ-M_ϕ alternating sequence in Algo. 1.

3.2 Application on Unsupervised Visual Tracking

Visual object tracking is a classic self–cycle-consistent task, with its cyclic structure illustrated in Fig. 1(c). Based on the paradigm proposed in this work, we can design two self-supervised schemes.

For a video sequence, let $\mathcal{X}$ and $\mathcal{Y}$ be the selected template and search frames, with $\mathbf{x}$ as a random object box in $\mathcal{X}$. Then, the loss function of the tracker T can be derived from Eq. 16:

$$\begin{aligned} &\mathcal{L}(T, \mathbf{x}) = \mathcal{L}_b(T(T(\mathbf{x}, \mathcal{X}, \mathcal{Y}), \mathcal{Y}, \mathcal{X})) + \mathcal{L}_b(T(\mathbf{x}, \mathcal{X}, \mathcal{Y}), \tilde{\mathbf{y}}), \\ &s.t. \quad \tilde{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathrm{BOX}_\mathcal{Y}} \mathrm{IoU}(\mathbf{y}, T(\mathbf{x}, \mathcal{X}, \mathcal{Y})), \end{aligned} \tag{17}$$

where $\mathcal{L}_b$ is the bounding-box loss, IoU represents the Intersection over Union between two bounding boxes, and $\mathrm{BOX}_\mathcal{Y}$ is detector-generated bounding-box set in frame $\mathcal{Y}$. The correspondence between this loss function and Eq.16 is as follows:

- $D_{cyc}(\mathbf{x}, g_\theta(g_\theta(\mathbf{x}, \mathcal{X}, \mathcal{Y}), \mathcal{Y}, \mathcal{X})) \Rightarrow \mathcal{L}_b(T(T(\mathbf{x}, \mathcal{X}, \mathcal{Y}), \mathcal{Y}, \mathcal{X}))$;
- $D_{sim}(g_\theta(\mathbf{x}, \mathcal{X}, \mathcal{Y}), \mathcal{Y}) \Rightarrow \mathcal{L}_b(T(\mathbf{x}, \mathcal{X}, \mathcal{Y}), \tilde{\mathbf{y}})$.

Using Algo. 2 as a reference, we propose an EM variant that bypasses $\mathbf{y}$ distribution estimation, where the expectation step computes $\hat{\mathbf{y}} = T(\mathbf{x}, \mathcal{X}, \mathcal{Y})$, and the maximization step updates T by optimizing the objective $\mathcal{L}_b(T(\hat{\mathbf{y}}, \mathcal{Y}, \mathcal{X}))$.

Since current trackers fail to achieve differentiable head-to-tail connections, we are compelled to develop CycleTrack from scratch. We map template boxes $\mathbf{x}$ to positional tokens via MLP, then concatenate them with uncropped frame tokens as input. The main architecture combines a vanilla ViT(Dosovitskiy et al., 2021) encoder, STARK(Yan et al., 2021)'s feature enhancer, and parallel FCOS(Tian et al., 2019) heads generating confidence-weighted box outputs. The tracker can be formally expressed as $\hat{\mathbf{y}} = T_\theta(\mathbf{x}, \mathcal{X}, \mathcal{Y})$. For clarity, we refer to the single-step trained tracker as CycleTrack and the EM-trained tracker as CycleTrack-EM.

3.3 Experiments

Table 5: Comparison with leading unsupervised trackers on LaSOT and TrackingNet.

Method	LaSOT		TrackingNet	
	AUC	Precision	AUC	Precision
ResPUL	-	-	54.6	48.5
LUDT+	30.5	28.8	56.3	49.5
USOT*	35.8	34.0	61.5	56.6
ULAST*-off	46.8	44.8	64.9	58.5
ULAST*-on	47.1	45.1	65.4	59.2
CycleTrack	51.0	49.7	75.9	71.5
CycleTrack-EM	**56.5**	**57.9**	**77.3**	**74.4**

Table 6: Comparison with leading strictly-unsupervised trackers on two datasets.

Method	LaSOT		TrackingNet	
	AUC	Precision	AUC	Precision
ResPUL	-	-	54.6	48.5
LUDT	26.2	23.4	54.3	46.9
USOT	33.7	32.3	59.9	55.1
ULAST-off	42.9	40.5	-	-
ULAST-on	43.3	40.7	-	-
CycleTrack	45.0	42.2	65.6	59.0
CycleTrack-EM	**51.2**	**49.9**	**69.1**	**64.7**

In CycleTrack and CycleTrack-EM, $\mathcal{L}_b$ adopts the weighted L1 and GIoU(Rezatofighi et al., 2019) losses consistent with STARK(Yan et al., 2021). The DETA detector(Ouyang-Zhang et al., 2022) produces detections on TrackingNet(Muller et al., 2018), GOT-10k(Huang et al., 2019), and LaSOT(Fan et al., 2019), combined with COCO(Lin et al., 2014) for unsupervised training. Optical flow labels are generated by ARFlow(Liu et al., 2020) on YouTube-VOS(Xu et al., 2018), ImageNet-ViD(Deng et al., 2009), GOT-10k, and LaSOT, serving as the strictly unsupervised set. We conduct comparisons with leading unsupervised trackers, including ResPUL(Wu et al., 2021), LUDT(Wang et al., 2021), USOT(Zheng et al., 2021), ULAST(Shen et al., 2022). We evaluate our unsupervised object tracking methods on LaSOT and TrackingNet, demonstrating significant advantages over existing approaches. Unlike conventional methods requiring pseudo-labels for image cropping, our tracker directly implements the proposed cyclic learning paradigm through full-image processing. Experimental results in Tab. 5 and 6 show our single-step and EM-based training approaches achieve state-of-the-art performance in both unsupervised (detector-annotated) and strictly unsupervised (optical-flow-annotated) settings, outperforming the second-best methods by considerable margins. Additionally, our framework naturally supports semi-supervised training (see Appendix for more details).

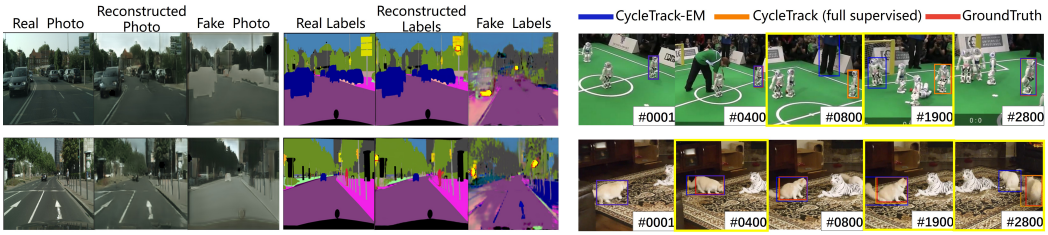
4 Further Analysis and Discussion

This section outlines the primary concerns of cyclic learning in applied contexts.

1. Is the mapping learned by cyclic learning always what we need?

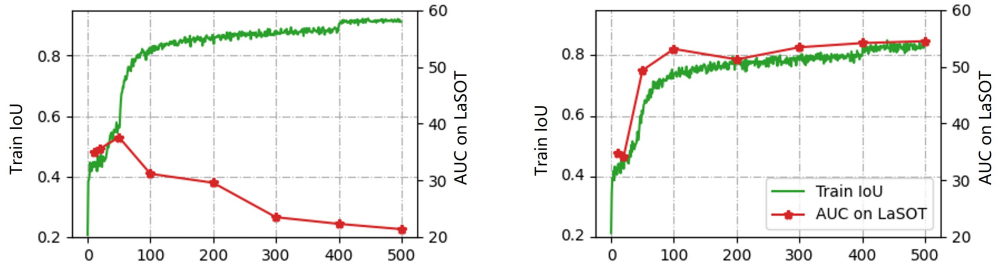
Not necessarily. We observe that cyclic learning models can learn incorrect mappings across paired domains. As shown in Fig. 4, although the reconstructed photos can closely match the real photos (and similarly for the reconstructed maps and the real maps), the intermediate fake images remain unsatisfactory. In visual tracking, CycleTrack occasionally locates incorrect objects in search frames. Owing to the annotation bias of the detector, the tracker retains the robustness to probabilistically re-localize the target even when initialized with random objects. This may cause the tracker to degenerate into a basic object detector. We attribute these failure cases primarily to the inherent limitations of cyclic learning itself. Without paired annotations, the models only learn some mapping between domains $\mathcal{X}$ and $\mathcal{Y}$ that satisfies cycle consistency but does not guarantee that the learned mapping is exactly what we intend. Introducing additional constraints beyond cycle consistency may help mitigate this issue.

Additionally, the EM method has an intrinsic risk of converging to local optima. During training, g_θ may gradually adapt to two distinct modes: $g_\theta(\mathbf{a}) = \mathbf{x}$ and $g_\theta(\mathbf{y}) = \mathbf{b}$. This means it simultaneously learns mappings from $\mathcal{A} \to \mathcal{X}$ and $\mathcal{Y} \to \mathcal{B}$, both of which can be achieved with a single set of



(a) Failure cases in CycleGN (b) Failure cases in CycleTrack-EM

Figure 4: Failure cases in cyclic learning. (a) CycleGN failures in photo↔map translation: While reconstructed images closely resemble real images, the generated fake images exhibit significant quality degradation. (b) Mislocalizations in CycleTrack-EM (highlighted by yellow boxes): Target objects are confused with other salient objects or visually similar distractors.



(a) Train CycleTrack via D_{cyc} (b) Train CycleTrack-EM via D_{cyc}

Figure 5: Ablation on D_{sim} removal (E-step removal): (a) Train tracker solely with D_{cyc}; (b) Freeze forward process via E-step, then train tracker with D_{cyc}.

parameters θ, where $\mathcal{A} = \{\mathbf{a}^i\}$ and $\mathcal{B} = \{\mathbf{b}^i\}$ represent domains corresponding to local optima. Correspondingly, f_ϕ learns the mappings $\mathcal{X} \to \mathcal{A}$ and $\mathcal{B} \to \mathcal{Y}$. As a result, we obtain a pair of bidirectional mappings: $\mathcal{X} \leftrightarrow \mathcal{A}$ and $\mathcal{B} \leftrightarrow \mathcal{Y}$, which deviate from the intended behavior $\mathcal{X} \leftrightarrow \mathcal{Y}$. Once a steady state where both cycles are satisfied is reached, the assumption $p_{data}(\mathbf{y}|\mathbf{x}) = q_\phi(\mathbf{y}|\mathbf{x})$ in an inner loop can never be achieved.

2. Which is better - single-step training or EM training?

In image translation, the single-step CycleGAN outperforms the EM-based CycleGN. Conversely, in visual tracking, the EM-based method CycleTrack-EM surpasses its single-step loss variant, CycleTrack. Besides, as introduced in the opening section, image generation tasks(Kwon & Park, 2019; Yang et al., 2020) tend to favor single-step optimization, whereas the REC-REG cycle(Yue et al., 2024; Wang et al., 2024) exhibits a preference for EM optimization. We argue that the choice between these two methods primarily depends on whether D_{KL} is well approximated. Both methods must employ the cycle-consistency loss D_{cyc}, which can be disregarded in this comparison. The accuracy of D_{sim} in measuring the KL divergence between sample and target distributions critically influences convergence behavior. For instance, while EM methods introduce inherent training instability, the GAN discriminator provides a robust D_{KL} surrogate that makes single-step optimization preferable.In contrast, aligning with the nearest detection box in CycleTrack is unreliable, because targets may remain completely undetected. This leads to poor D_{sim} estimation, explaining CycleTrack's inferior performance compared to CycleTrack-EM. An alternative perspective is that, compared to paired cyclic tasks, self-cyclic tasks facilitate more stable EM training by eliminating the need to wait for one EM process to converge before initiating the next. We provide a more detailed analysis of the applicable scenarios in Appendix C.

3. Can we rely solely on the cycle-consistency loss?

Using only the D_{cyc} loss in single-step optimization is mathematically equivalent to removing the E-step in the EM algorithm by eliminating the stop-gradient operation, which introduces a key limitation: the generated $\hat{y}$ often fails to adhere to the target domain or distribution due to the lack of

explicit constraints. In our implementation, we disable the E-step by removing the forward-process freezing operation in CycleTrack-EM. As evidenced by Fig. 5, this ablated model converges to clearly trivial solutions compared to the complete EM training procedure—a behavior that parallels the significant performance deterioration observed in CycleGAN when removing its adversarial loss while retaining only the cycle-consistency loss. This validates the necessity of both the D_{sim} loss for single-step methods and the E-step for EM algorithms.

5 Conclusion

This work introduces a novel probabilistic framework that unifies cycle-consistent learning through variational modeling. By formulating cyclic tasks within a general theoretical foundation, we establish principled connections between previously disparate approaches. The framework naturally gives rise to two complementary optimization strategies - a VAE-style single-step method for efficient training, and an EM variant that operates without KL divergence estimation. Together, these contributions provide both theoretical coherence and practical flexibility for cycle-consistent learning across tasks. In image translation, our framework theoretically explains CycleGAN's success as variational approximation of cycle-consistent learning, while enabling GAN-free alternatives via EM optimization. For object tracking, self-cyclic constraints dynamically model appearance variations, where both single-step and EM-based trackers achieve robust unsupervised performance. We hope this work will inspire future research directions in cyclic learning.

References

Amjad Almahairi, Sai Rajeshwar, Alessandro Sordoni, Philip Bachman, and Aaron Courville. Augmented cyclegan: Learning many-to-many mappings from unpaired data. In *International conference on machine learning*, pp. 195–204. PMLR, 2018.

Peijun Bao, Zihao Shao, Wenhan Yang, Boon Poh Ng, and Alex C Kot. E3m: zero-shot spatio-temporal video grounding with expectation-maximization multimodal modulation. In *European Conference on Computer Vision*, pp. 227–243. Springer, 2024.

Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6182–6191, 2019.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15750–15758, 2021.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13608–13618, 2022.

J. Deng, W. Dong, R. Socher, L. Li, L. Kai, and F. F Li. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1801–1810, 2019.

Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5374–5383, 2019.

Shenyuan Gao, Chunluan Zhou, and Jun Zhang. Generalized relation modeling for transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18686–18695, June 2023.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

Nilesh Kulkarni, Abhinav Gupta, and Shubham Tulsiani. Canonical surface mapping via geometric cycle consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2202–2211, 2019.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Yong-Hoon Kwon and Min-Gyu Park. Predicting future frames using retrospective cycle gan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1811–1820, 2019.

Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 2002.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

L. Liu, J. Zhang, R. He, Y. Liu, Y. Wang, Y. Tai, D. Luo, C. Wang, J. Li, and F. Huang. Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6488–6497, Los Alamitos, CA, USA, jun 2020. IEEE Computer Society.

Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. *Advances in neural information processing systems*, 29, 2016.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 300–317, 2018.

Radford M Neal and Geoffrey E Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pp. 355–368. Springer, 1998.

Jeffrey Ouyang-Zhang, Jang Hyun Cho, Xingyi Zhou, and Philipp Krähenbühl. Nms strikes back. *arXiv preprint arXiv:2212.06137*, 2022.

Meng Qu, Yoshua Bengio, and Jian Tang. Gmnn: Graph markov neural networks. In *International conference on machine learning*, pp. 5241–5250. PMLR, 2019.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 658–666, 2019.

Qiuhong Shen, Lei Qiao, Jinyang Guo, Peixia Li, Xin Li, Bo Li, Weitao Feng, Weihao Gan, Wei Wu, and Wanli Ouyang. Unsupervised learning of accurate siamese tracking. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8091–8100, 2022. doi: 10.1109/CVPR52688.2022.00793.

Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2107–2116, 2017.

Zhiqing Sun and Yiming Yang. An em approach to non-autoregressive conditional sequence generation. In *International Conference on Machine Learning*, pp. 9249–9258. PMLR, 2020.

Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9627–9636, 2019.

Leonid Nisonovich Vaserstein. Markov processes over denumerable products of spaces, describing large systems of automata. *Problemy Peredachi Informatsii*, 5(3):64–72, 1969.

Ning Wang, Yibing Song, Chao Ma, Wengang Zhou, Wei Liu, and Houqiang Li. Unsupervised deep tracking. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1308–1317, 2019a. doi: 10.1109/CVPR.2019.00140.

Ning Wang, Wengang Zhou, Yibing Song, Chao Ma, Wei Liu, and Houqiang Li. Unsupervised deep representation learning for real-time tracking. *International Journal of Computer Vision*, 129: 400–418, 2021.

Ning Wang, Jiajun Deng, and Mingbo Jia. Cycle-consistency learning for captioning and grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 5535–5543, 2024.

Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2566–2576, 2019b.

Qiangqiang Wu, Jia Wan, and Antoni B. Chan. Progressive unsupervised learning for visual object tracking. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2992–3001, 2021. doi: 10.1109/CVPR46437.2021.00301.

Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision – ECCV 2018*, pp. 603–619, Cham, 2018. Springer International Publishing.

Sihan Xu, Ziqiao Ma, Yidong Huang, Honglak Lee, and Joyce Chai. Cyclenet: Rethinking cycle consistency in text-guided diffusion for image manipulation. *Advances in Neural Information Processing Systems*, 36:10359–10384, 2023.

Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. *arXiv preprint arXiv:2103.17154*, 2021.

Heran Yang, Jian Sun, Aaron Carass, Can Zhao, Junghoon Lee, Jerry L Prince, and Zongben Xu. Unsupervised mr-to-ct synthesis using structure-constrained cyclegan. *IEEE transactions on medical imaging*, 39(12):4249–4261, 2020.

Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pp. 341–357. Springer, 2022.

Weihao Yuan, Michael Yu Wang, and Qifeng Chen. Self-supervised object tracking with cycle-consistent siamese networks. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10351–10358, 2020. doi: 10.1109/IROS45743.2020.9341621.

Tongtian Yue, Jie Cheng, Longteng Guo, Xingyuan Dai, Zijia Zhao, Xingjian He, Gang Xiong, Yisheng Lv, and Jing Liu. Sc-tune: Unleashing self-consistent referential comprehension in large vision language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13073–13083, 2024.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pp. 649–666. Springer, 2016.

Jilai Zheng, Chao Ma, Houwen Peng, and Xiaokang Yang. Learning to track objects from unlabeled videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13546–13555, October 2021.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

A Extended Inferences

A.1 From maximizing log-likelihood to maximizing ELBO.

Here, we derive Eq. 3 in detail, and similar steps are followed for the ELBO-related parts in the paper:

$$\begin{aligned}
\log p_\theta(\mathbf{x}) =& \log \int q_\phi(\mathbf{y}|\mathbf{x}) p_\theta(\mathbf{x}) d\mathbf{y} \\
\geq& \int q_\phi(\mathbf{y}|\mathbf{x}) \log p_\theta(\mathbf{x}) d\mathbf{y} \\
=& \int q_\phi(\mathbf{y}|\mathbf{x}) \log \frac{p_\theta(\mathbf{x},\mathbf{y})}{p_{data}(\mathbf{y}|\mathbf{x})} d\mathbf{y} \\
=& \int q_\phi(\mathbf{y}|\mathbf{x}) \log \left(\frac{p_\theta(\mathbf{x},\mathbf{y})}{q_\phi(\mathbf{y}|\mathbf{x})} \cdot \frac{q_\phi(\mathbf{y}|\mathbf{x})}{p_{data}(\mathbf{y}|\mathbf{x})} \right) d\mathbf{y} \\
=& \int q_\phi(\mathbf{y}|\mathbf{x}) \log \frac{p_\theta(\mathbf{x},\mathbf{y})}{q_\phi(\mathbf{y}|\mathbf{x})} d\mathbf{y} + \int q_\phi(\mathbf{y}|\mathbf{x}) \log \frac{q_\phi(\mathbf{y}|\mathbf{x})}{p_{data}(\mathbf{y}|\mathbf{x})} d\mathbf{y} \\
=& \mathbb{E}_{q_\phi(\mathbf{y}|\mathbf{x})} \left[\log \frac{p_\theta(\mathbf{x},\mathbf{y})}{q_\phi(\mathbf{y}|\mathbf{x})} \right] + D_{KL}(q_\phi(\mathbf{y}|\mathbf{x}) || p_{data}(\mathbf{y}|\mathbf{x})) \\
\geq& \ell_{\theta,\phi}(\mathbf{x}),
\end{aligned} \tag{18}$$

with

$$\begin{aligned}
\ell_{\theta,\phi}(\mathbf{x}) =& \mathbb{E}_{q_\phi(\mathbf{y}|\mathbf{x})} \left[\log \frac{p_\theta(\mathbf{x},\mathbf{y})}{q_\phi(\mathbf{y}|\mathbf{x})} \right] \\
=& \int q_\phi(\mathbf{y}|\mathbf{x}) \log \frac{p_\theta(\mathbf{x}|\mathbf{y}) p_{data}(\mathbf{y})}{q_\phi(\mathbf{y}|\mathbf{x})} d\mathbf{y} \\
=& \int q_\phi(\mathbf{y}|\mathbf{x}) \log p_\theta(\mathbf{x}|\mathbf{y}) d\mathbf{y} + \int q_\phi(\mathbf{y}|\mathbf{x}) \log \frac{p_{data}(\mathbf{y})}{q_\phi(\mathbf{y}|\mathbf{x})} d\mathbf{y} \\
=& \mathbb{E}_{q_\phi(\mathbf{y}|\mathbf{x})} \left[\log p_\theta(\mathbf{x}|\mathbf{y}) \right] - D_{KL}(q_\phi(\mathbf{y}|\mathbf{x}) || p_{data}(\mathbf{y})).
\end{aligned} \tag{19}$$

A.2 Another constraint: feature consistency.

The feature consistency assumes that corresponding data points from the two domains can overlap when mapped to the same space, or that paired data samples share domain-invariant features. In essence, this serves as a way to constrain $\mathbf{y}$ by aligning $\mathbf{x}$ and $\mathbf{y} = f(\mathbf{x})$ in a particular feature space, thereby replacing or enhancing the role of $D_{KL}(q_\phi(\mathbf{y}|\mathbf{x})||p_{data}(\mathbf{y}))$ in single-step optimization. The difference between feature consistency and cycle consistency in constraining the range of $\mathbf{y}$ lies in their metrics: the former is represented by $D_{sim}(\mathbf{y}, \mathbf{x})$ while the latter is approximated by $D_{sim}(\mathbf{y}, \mathcal{Y})$. Since additional reasonable constraints are introduced, the mapping quality may improve as mentioned in the main text, but $D_{sim}(\mathbf{y}, \mathbf{x})$ may also lack proper definition in practical applications like $D_{sim}(\mathbf{y}, \mathcal{Y})$.

For instance, Wang et al. (2019b) models the relationship between forward and backward pixels as a self-cycle-consistent task, performing cyclic training across multiple frames with a one-step training pattern. In Eq. 16, D_{cyc} is expressed in this work as the requirement for pixel alignment within the cycle formed by different frames, while D_{sim} enforces feature consistency for the object across frames, serving as a replacement for the KL divergence. Dwibedi et al. (2019) proposes a self-supervised representation learning method which aims to learn frame-level embeddings by temporally aligning video sequences. This task is formulated as a *self-consistent task*. The method enforces D_{cyc} by constructing bidirectional mappings between video frames. Meanwhile, D_{KL} is implicitly satisfied as the same encoder maps inputs into the embedding space. Additionally, the mapping process in the embedding space inherently enforces a feature consistency constraint.

Since feature consistency falls outside the scope of this work, we provide only a brief introduction here.

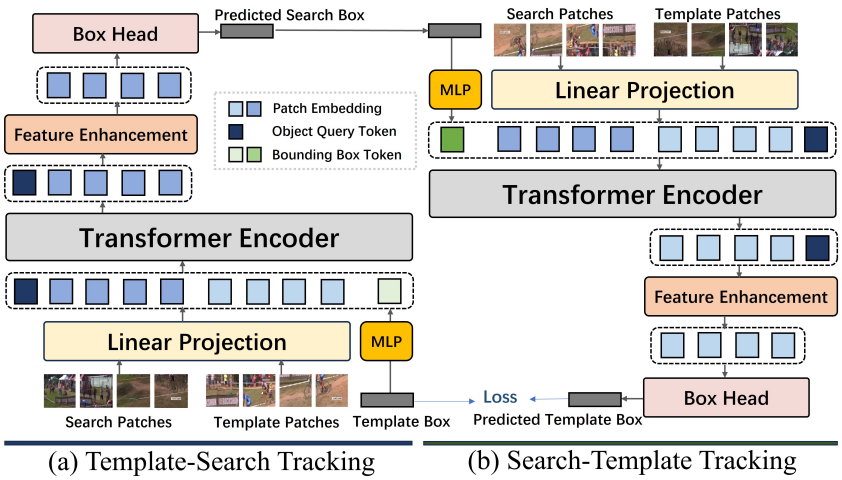


Figure 6: The structure of CycleTrack.

Table 7: The performance of different optimization methods in image translation tasks.

Loss	Per-pixel accuracy	Per-class accuracy	Class IOU
$Labels \rightarrow Photo$			
$D_{cyc}^{\mathcal{X}} + D_{sim}^{\mathcal{X}}$	0.41	0.07	0.03
$D_{cyc}^{\mathcal{Y}} + D_{sim}^{\mathcal{Y}}$	0.39	0.05	0.02
$D_{cyc}^{\mathcal{X}} + D_{sim}^{\mathcal{X}} \rightleftarrows D_{cyc}^{\mathcal{Y}} + D_{sim}^{\mathcal{Y}}$	0.51	0.13	0.10
CycleGN	0.52	0.14	0.10
CycleGAN(Zhu et al., 2017)	0.52	0.17	0.11
$Photo \rightarrow Labels$			
$D_{cyc}^{\mathcal{X}} + D_{sim}^{\mathcal{X}}$	0.10	0.06	0.02
$D_{cyc}^{\mathcal{Y}} + D_{sim}^{\mathcal{Y}}$	0.32	0.09	0.05
$D_{cyc}^{\mathcal{X}} + D_{sim}^{\mathcal{X}} \rightleftarrows D_{cyc}^{\mathcal{Y}} + D_{sim}^{\mathcal{Y}}$	0.52	0.16	0.11
CycleGN	0.51	0.16	0.10
CycleGAN(Zhu et al., 2017)	0.58	0.22	0.16

B Extended Experiments

B.1 Several variants of single-step loss.

This paper focuses on the joint optimization of $\max(\log p(\mathbf{x}) + \log p_\phi(\mathbf{y}))$. We show that optimizing only a single direction, specifically maximizing $p(\mathbf{x})$ alone or $p(\mathbf{y})$ alone, influences performance in both forward and backward tasks. Tab. 7 summarizes the results, where $D_{cyc}^{\mathcal{X}} + D_{sim}^{\mathcal{X}}$ and $D_{cyc}^{\mathcal{Y}} + D_{sim}^{\mathcal{Y}}$ correspond to maximizing $\log p(\mathbf{x})$ and $\log p(\mathbf{y})$ respectively. Another variant involves rotated optimization of $\log p(\mathbf{x})$ and $\log p(\mathbf{y})$, denoted in the table as $D_{cyc}^{\mathcal{X}} + D_{\text{sim}}^{\mathcal{X}} \rightleftarrows D_{cyc}^{\mathcal{Y}} + D_{sim}^{\mathcal{Y}}$. Unlike Algo. 1, this approach switches objectives every mini-batch without awaiting convergence between two EM cycles. While this configuration demonstrates inferior performance to CycleGAN on both tasks, it remains viable. We argue that under constrained training conditions, splitting Equation 8 into dual cyclic objectives with rotated training provides a feasible alternative.

B.2 Fully-Supervised and Semi-supervised Experiments of CycleTrack.

CycleTrack was evaluated under identical settings to STARK for supervised training. In the semi-supervised setup, only the annotation of the target in the first frame of each sequence was available. Since the similarity loss $D_{sim}(\hat{\mathbf{y}}, \mathcal{Y})$ cannot be defined in search frames under this configuration, the single-step optimization method becomes inapplicable. Therefore, we exclusively employed the

Table 8: Comparison between fully-supervised and semi-supervised CycleTrack with state-of-the-art fully-supervised trackers on LaSOT and TrackingNet.

Method	LaSOT		TrackingNet	
	AUC	Precision	AUC	Precision
DiMP(Bhat et al., 2019)	56.9	56.7	74.0	68.7
STARK(Yan et al., 2021)	67.1	-	82.0	-
Mixformer-22k(Cui et al., 2022)	69.2	74.7	<u>83.1</u>	81.6
OSTrack(Ye et al., 2022)	69.1	75.2	<u>83.1</u>	<u>82.0</u>
GRM(Gao et al., 2023)	<u>69.9</u>	**75.8**	**84.0**	**83.3**
CycleTrack (fully)	**70.1**	<u>75.2</u>	82.9	81.0
CycleTrack-EM (semi)	63.6	66.7	80.6	78.3

EM approach for semi-supervised training. As shown in Tab. 8, CycleTrack achieves comparable performance to leading two-frame trackers, demonstrating that our assembled network possesses the fundamental capabilities expected of a competent tracker. Notably, the semi-supervised CycleTrack-EM delivers remarkably strong performance - showing minimal degradation compared to its fully-supervised counterpart and even surpassing the fully-supervised DiMP. These results validate the effectiveness of our proposed training methodology.

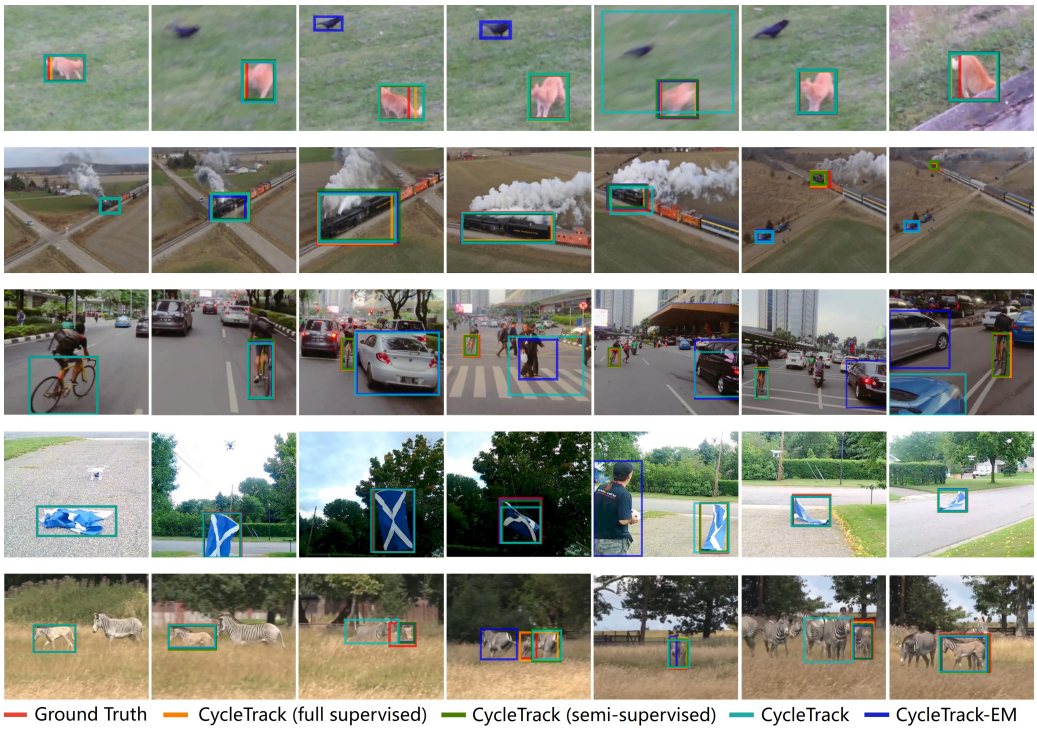
C Extended Discussions

Given that the single-step approach is more conceptually straightforward and has achieved remarkable success in CycleGAN-related applications, while the EM-based method is less prevalent, we provide the following recommendations for its applicable scenarios:

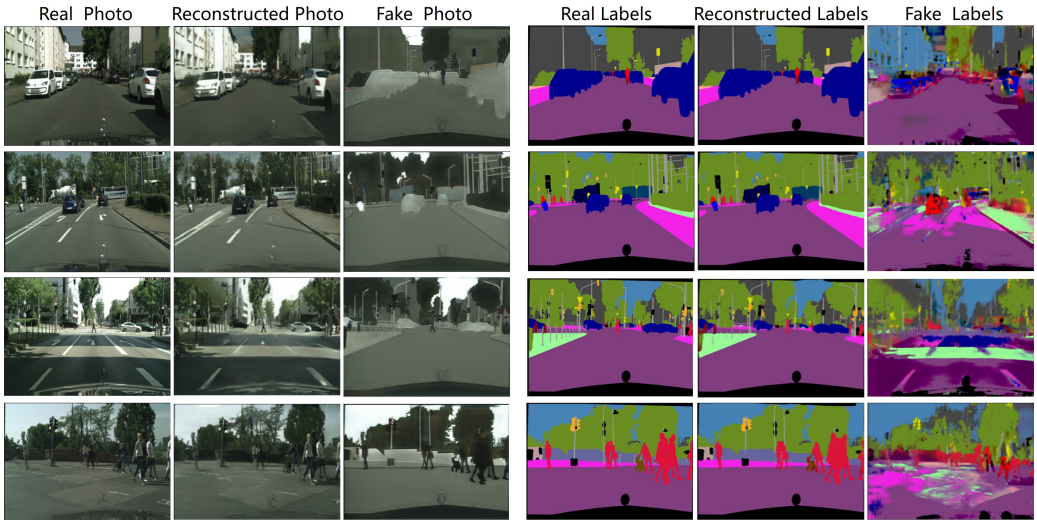
i. Scenarios without D_{KL} estimation. GANs serve as an effective estimator of D_{KL}, but they are not universally suitable for all tasks. In practice, apart from GANs, few methods can reliably approximate KL divergence. While image-based domain adaptation largely adopts CycleGAN and uses GANs as the estimator, other modalities attempting to use single-step losses have generally achieved suboptimal results. This has compelled some studies to resort to alternatives such as feature consistency(discussed in Appendix A.2). Adopting EM-based methods may offer a new perspective for addressing problems in non-GAN scenarios.

ii. Scenarios where convergence to local optima can be avoided. a). Semi-supervised settings with limited annotated data: Even a small amount of annotated data can prevent domains $\mathcal{X}$ and $\mathcal{Y}$ from forming independent cycles. In such configurations, domains $\mathcal{X}$ and $\mathcal{Y}$ are necessarily connected. Even if the network tends to generate simpler outputs, the locally optimal domains $\mathcal{B}$ and $\mathcal{A}$ will remain closer to the true domains $\mathcal{X}$ and $\mathcal{Y}$. **b).** When f_ϕ and g_θ are already capable of generating outputs in the target domain: In this case, although the network lacks direct mapping ability between domains, it has acquired the capacity to generate results in domain $\mathcal{Y}$ through other upstream tasks (even if not starting from domain $\mathcal{X}$). EM training then acts as fine-tuning on a pre-trained network, significantly reducing the possibility of the network freely diverging to other domains. **c).** When no alternative domains are available to the generative function—i.e., in strictly bijective scenarios: While training CycleTrack, we observed that for any output quadruple beyond object bounding boxes, there are hardly any other meaningful or structured combinations that can satisfy cycle consistency. This makes producing object bounding descriptions the most straightforward choice for the network.

iii. Scenarios that call for a quick and simple trial: In practice, EM training involves only the reconstruction loss D_{cyc}, which is generally the simplest loss to design across tasks and modalities. In contrast, designing and optimizing D_{sim} is considerably more challenging. Another key consideration lies in the structural constraints imposed by single-step loss training: for f_ϕ and g_θ to be trained end-to-end, they must form a fully differentiable pipeline. Satisfying this requirement often places strong constraints on the architecture of both networks—sometimes even necessitating customized designs, as encountered in our work on CycleTrack. In contrast, the EM approach imposes no structural constraints on f_ϕ and g_θ, since the two networks are trained separately. This enables the use of non-differentiable operations during training.



(a) Visualization of tracking results by different CycleTrack variants on LaSOT.



(b) Visualization of photo↔labels translation results by CycleGN on Cityscape.