OPEL

Optimal Transport Guided ProcedurE Learning

Under the guidance of Prof. Ashutosh Modi, CSE, IITK

Presented by:

Aritra Ambudh Dutta
Ashish Upadhyay
Siddhant Shekhar

Introduction to OPEL

Optimal Transport Guided Procedure Learning (OPEL) is a framework for inferring key procedural steps and their temporal order from instructional videos by formulating step alignment as an Optimal Transport (OT) problem. Unlike prior methods relying on rigid or monotonic alignments, OPEL introduces regularizations that promote temporally coherent and semantically meaningful mappings, achieving significant improvements across multiple benchmarks.

We shall, in short, delve deeper into Optimal Transport before presenting the Actual Paper.

History of Optimal Transport

01. Monge Problem

02. Kantorovich Problem

03. The Dual Problem

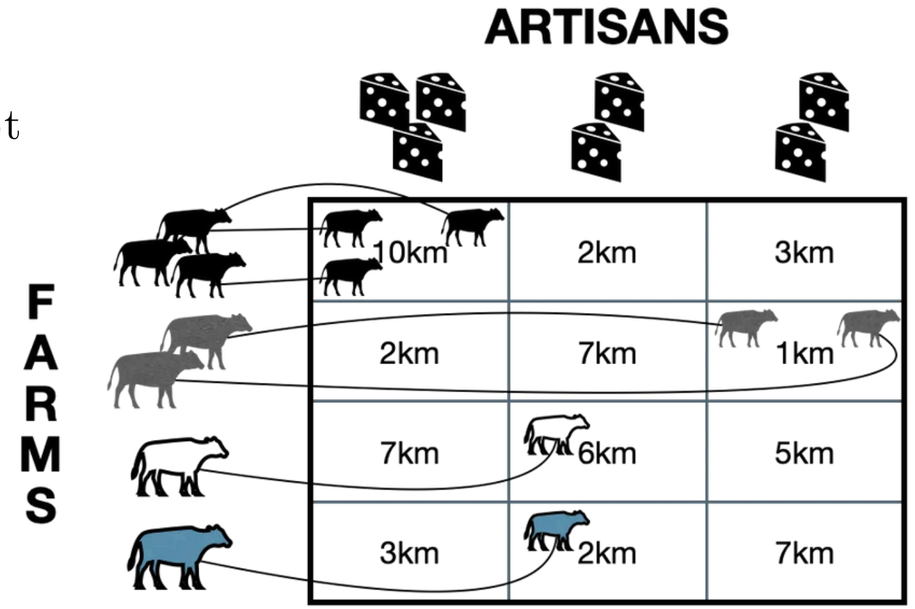
Monge Problem

Intuition

- Suppose we have some sets of cows and we have to transport each set to different locations of artisans and there is some cost which is incurred for moving a set of cows to any locations

- We have make a transport plan to make the total cost of moving the cows to artisans.

- There may not always exist a solution if the distribution is not absolutely continuous.
- It is a non-convex and difficult to solve directly.

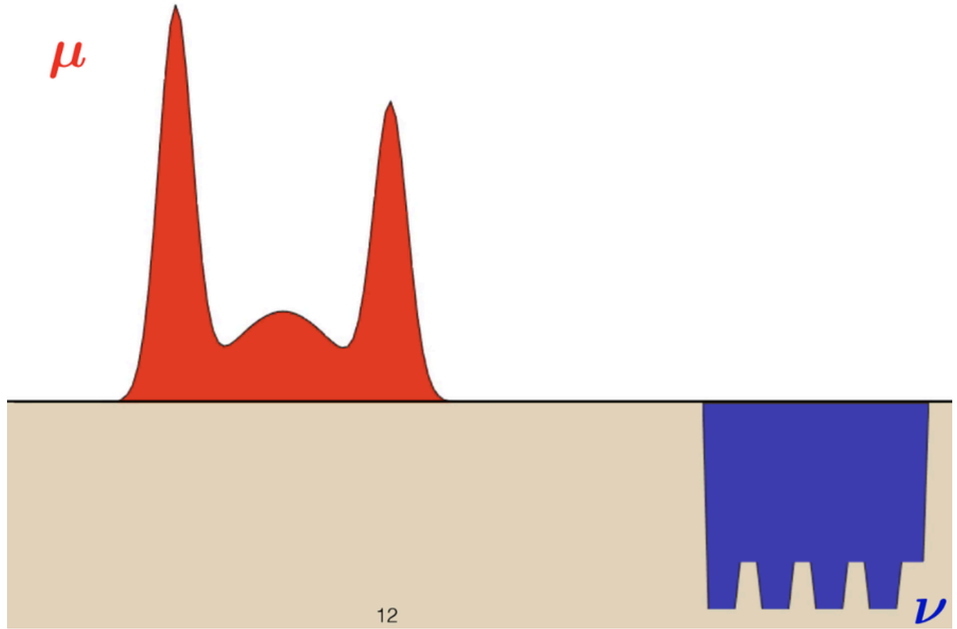


Monge Problem

Mathematical Explanation

On a given Probability Space X:

- A source distribution of probability measure μ
- A target distribution of probability measure ν
- A cost function c(x,y): X × X → ℝ, cost of transporting one unit from x to y; usually the distance or squared distance between points x and y.



Goal:
Find a transfer map π that moves every point x in μ to a point π(x) in ν, such that when the pushforward of μ under π equals ν, i.e., mass is preserved, the total transportation cost is minimized, i.e.,

$$\min_{\pi \in \Pi(\mu,\nu)} \int_{X \times X} c(x, \pi(x)) d\mu$$

Note:
π is one-to-one and for a set S ⊂ X, we have that μ(S) = ν(π(S)). Π(μ, ν) denotes the set of all transfer maps from μ to ν

Monge Problem

Limitations

While the physical interpretation of such a plan can be seen, actually finding such a plan remains a difficult problem to this day. The cases where $c(x, y) = |x - y|$ and $c(x, y) = (x - y)^2$ are fairly well studied and can be solved, but more exotic cost functions are less tractable.

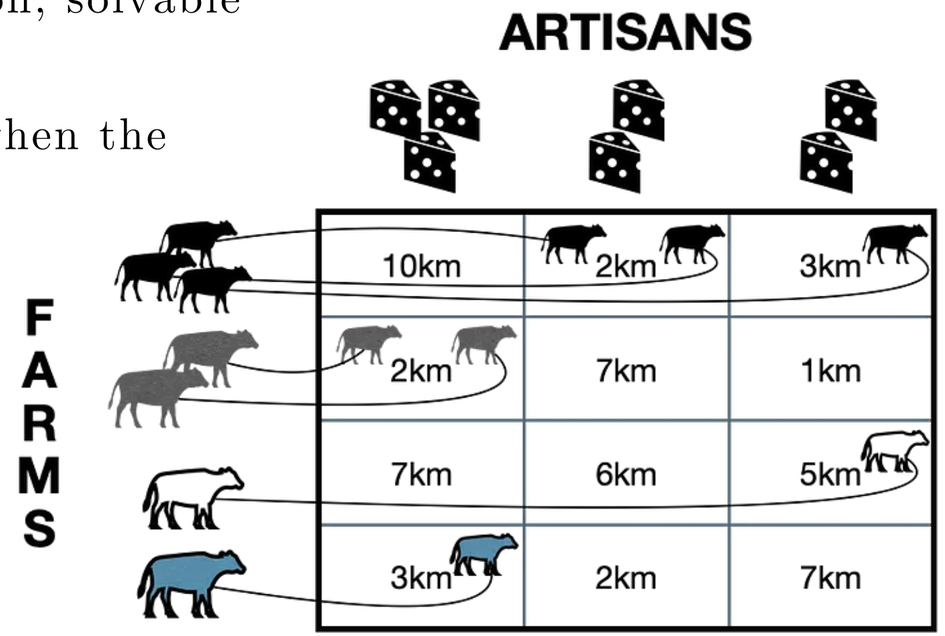
Furthermore, the Monge problem may be ill-posed for certain distributions μ, ν. For example, for $X = \mathbb{R}$, $\mu = \delta_0$, $\nu = \frac{1}{2}\delta_1 + \frac{1}{2}\delta_2$ (where δ_k is the Dirac delta function centered at k), there are no admissible transport maps at all, because we cannot split the mass located at 0 to send it to 1 or 2, as such a map would not be one-to-one. Leonid Kantorovich later recognized that relaxing this one-to-one condition would make the problem significantly more tractable and allow for a linear programming approach to be used. We now present the modern formulation of the Monge-Kantorovich Problem.

Kantorovich Problem

Intuition

- Instead of rigid assignments shown in the Monge Problem, here we allow a more flexible plan: mass (Cows) can be split and transported to multiple locations (Artisans).
- Transport plan tells us what fraction of material goes from each source to each destination.

- This turns the problem into a linear (convex) optimization, solvable efficiently.
- Always admits a solution via linear programming, even when the distributions are continuous or uneven.

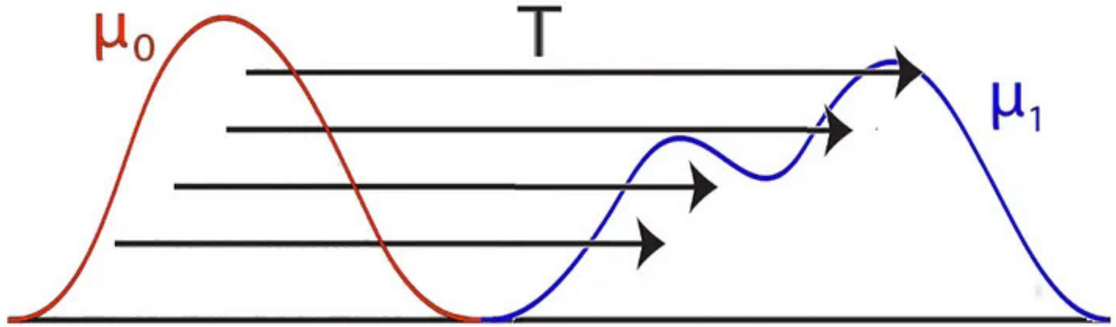


Monge-Kantorovich Problem

Let μ denote a probability measure on a space X, and let ν denote a probability measure on a space Y . Let c : X × Y → ℝ be a cost function. We say that π : X × Y → ℝ is a transference plan if it is a probability measure on X × Y , and for arbitrary sets S ⊂ X, T ⊂ Y , we have that π(S × Y) = μ(S) and π(X × T) = ν(T). We let Π(μ, ν) denote the set of all admissible transference plans. Then the problem is to minimize:

$$\min_{\pi \in \Pi(\mu,\nu)} \int_{X \times Y} c(x,y) d\pi$$

The central idea of the relaxation is the probability distribution, which changed from Monge's idea of a one-to-one function which described something like a matching, into a probability distribution over the combined spaces.



N. Papadakis, Optimal Transport for Image Processing, habilitation à diriger des recherches, Université de Bordeaux, Dec. 2015

The Dual Problem

Let $\psi : X \to \mathbb{R}$ and $\phi : Y \to \mathbb{R}$ be integrable functions such that for almost every $(x, y) \in X \times Y$ (outside a set of measure zero), $\psi(x) + \phi(y) \leq c(x, y)$. The dual problem is then the maximization:

$$\max_{\psi,\phi} \int_X \psi(x) d\mu + \int_Y \phi(y) d\nu.$$

As this is a dual problem, the two solutions in fact should coincide:

$$\inf_{\pi \in \Pi(\mu,\nu)} \int_{X \times Y} c(x, y) d\pi = \sup_{\psi,\phi} \left(\int_X \psi(x) d\mu + \int_Y \phi(y) d\nu \right)$$

Monge-Kantorovich Duality

The Dual Problem

Economic Intuition

Suppose that X represents a space of bakeries (with bakeries positioned according to their characteristics, such as clientele and location), and Y represents a space of cafes serving baked goods. Bakeries produce bread according to the probability distribution μ (so for a finite number of bakeries, this is a probability distribution on point masses), and cafes demand bread according to the probability distribution ν. Acting on their own, the bakeries and cafes must incur some cost c(x, y) to transport a unit of bread from bakery x to cafe y, and their problem is to find the distribution π such that the total cost of transporting bread, $\int_{X\times Y} c(x,y)d\pi$ is minimized.

Now, suppose a transportation company offers to take care of the transportation between bakeries and cafes. They charge a flat fee for pickup at a bakery: ψ(x) per unit, depending on the bakery x, and another fee for delivery to a cafe: ϕ(y) per unit, depending on the target cafe y. The company guarantees that their prices are competitive: $\psi(x) + \phi(y) \leqslant c(x, y)$ for almost every pair (x, y), so it is always worth the bakery/cafe's while to use the transportation company's services.

The Dual Problem

Economic Intuition

Then, the duality statement implies that for the transportation company, there is some pair of pricings ψ, ϕ such that the transportation company earns essentially as much as the bakeries/cafes were spending on their own. (So essentially all the slack in the market is taken up by the transportation company.)

Marginal Constraints (for arbitrary sets $S \subset X$, $T \subset Y$, $\pi(S \times Y) = \mu(S)$ and $\pi(X \times T) = \nu(T)$) ensure that:

1. We are not sending too many cakes from a bakery (Row sum for Bakery x in π must be equal to $\mu(x)$)
2. A cafe isn't getting too few cakes (Column sum for Cafe y in π must be equal to $\nu(y)$)

Under regularity conditions (e.g., μ has a density, $c(x, y) = |x-y|^2$), Brenier's theorem guarantees the existence of a unique optimal transport map $\boldsymbol{\pi}$.
π is the gradient of a convex function: $\pi(x) = \nabla\varphi(x)$ where $\varphi(x)$ is the Kantorovich potential. Solving the Primal Problem (Kantorovich) and the Dual Problem (Kantorovich Duality) will help us get the unique $\pi(x)$.

Optimal Transport

Optimal transport is a mathematical theory that studies the most efficient way to move or transform one distribution of mass (or probability) into another, minimizing a cost associated with the transportation.

This cost of transforming one probability distribution to another is done by calculating the dissimilarity between them using the Wasserstein distance (also called the Earth Mover's Distance, EMD)

Minimising the Wasserstein Distance using linear programming is computationally very expensive in higher dimensions.
For p=1, the convex problem can be easily solved using the duality property which can be solved easily by avoiding the coupling.

Theorem (Kantorovich duality)

$$\min_{\gamma \in M_+(\mathcal{X}\times\mathcal{Y})} \left\{ \int_{\mathcal{X}\times\mathcal{Y}} c(x,y)d\gamma(x,y) : \pi^x_{\#}\gamma = \mu,\ \pi^y_{\#}\gamma = \nu \right\} \quad \text{(P)}$$

$$=$$

$$\max_{\substack{\phi\in L^1(\mu)\\ \psi\in L^1(\nu)}} \left\{ \int_{\mathcal{X}} \phi(x)d\mu(x) + \int_{\mathcal{Y}} \psi(y)d\nu(y) : \phi(x)+\psi(y) \le c(x,y) \right\} \quad \text{(D)}$$

Wasserstein Distances

Let $p \geq 1$. Let $\boldsymbol{c}(x,y) := \boldsymbol{D}^p(x,y)$, a metric.

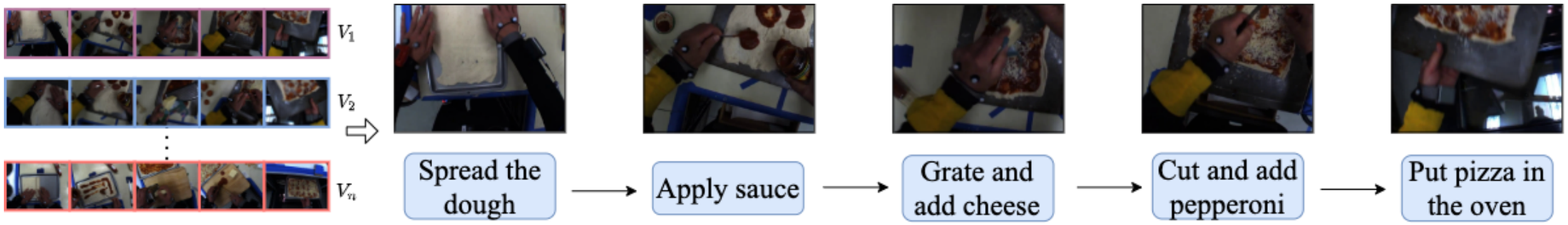
Def. The p-**Wasserstein distance** between $\boldsymbol{\mu}, \boldsymbol{\nu}$ in $\mathcal{P}(\Omega)$ is

$$W_p(\boldsymbol{\mu},\boldsymbol{\nu}) \overset{\text{def}}{=} \left(\inf_{\boldsymbol{P}\in\Pi(\boldsymbol{\mu},\boldsymbol{\nu})} \iint \boldsymbol{D}(x,y)^p \boldsymbol{P}(dx,dy) \right)^{1/p}.$$

Procedure Learning

Procedure learning represents a critical task in computer vision and artificial intelligence, focusing on the automatic identification of key steps and their logical sequencing from multiple video demonstrations of the same task. This challenge extends beyond simple action recognition, as it requires understanding *temporal relationships*, handling *variations in execution speed*, and managing *non-monotonic sequences* that naturally occur when different individuals perform the same procedure.

The complexity of this problem becomes evident when considering real-world scenarios where individuals may execute tasks in *slightly different orders*, *skip certain steps*, or *repeat actions*.



Procedure Learning

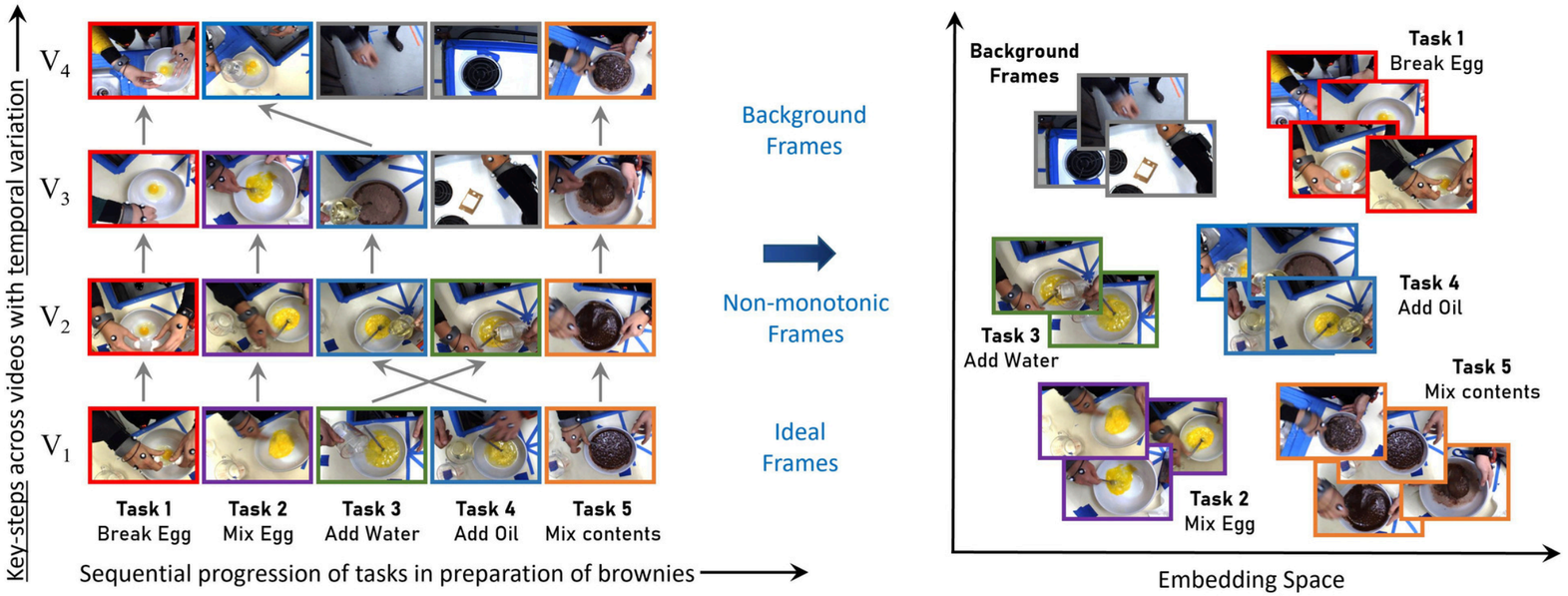
Pre-existing Works on PL

In PL, traditional approaches would necessitate hardcoding tedious explicit instructions for each sub-task of the process (thus difficult to scale and generalize). Action-based tasks focus on a single video and thus fail to identify repetitive key steps across multiple occurrences of the same task. They typically neglect the sequencing of events, crucial for discerning an overall expected procedure composed of the sub-tasks. If PL is done in a supervised setting, the reliance on per-frame annotations demands extensive manual labor. Similar scalability issues can arise when weakly supervised learning methodology is applied because one has to label all the subtasks by watching the videos, which is practically infeasible for large tasks like Sign language Processing. Recent studies have shifted the focus towards Self-Supervised learning, where the autonomous agent learns directly from observing multiple demonstrations of the assembly, without the need for any labels. But this assumes a complex task is unfolded into a *predictable* and *monotonic* manner *(i.e. consistent order of actions across sequences).*

However, real-world sequences frequently deviate from this pattern, exhibiting temporal non-uniformities as depicted in the Figure on the next page.

Procedure Learning

OPEL



Key steps required to prepare a brownie.

The sequences showcase temporal variations and corresponding key-step alignment challenges, namely (i) background frames (depicted as gray blocks), (ii) non-monotonic frames. OPEL aims to learn an embedding space where corresponding key-steps have similar embeddings while tackling the above challenges.

Procedure Learning

OPEL v/s State-of-the-art (SOTA) Methods

For both third-person and first-person (egocentric) videos, SOTA methods aim at finding correspondences across videos in time to accomplish procedure learning. However, to establish temporal relationships within the sequences, these methods often rely on *temporal alignment (frame-to-frame mapping)* or assume *monotonic alignment* of video pairs, leading to sub-optimal results. This is often due to the real-life deviations, as mentioned earlier. They can be classified into 3 categories: (i) *background frames*: frames irrelevant to the primary activity and should thus be excluded from alignment; (ii) *redundant frames*: these frames appear only in one sequence but not in others and do not contribute to the task; (iii) *non-monotonic frames*: these frames are characterized by a non-monotonic sequence of actions.

OPEL proposed to treat the video frames as samples from an unknown distribution, enabling to frame their distance calculation as an optimal transport (OT) problem. Notably, the OT-based formulation allows us to relax the previously mentioned assumptions. To further improve performance, we enhance the OT formulation by introducing two regularization terms, i.e., Inverse Difference Moment Regularization and K-L Divergence, *using exponentially decaying priors.*

Procedure Learning

Why OPEL?

Inverse Difference Moment Regularization

The first regularization term, inverse difference moment regularization, serves a dual purpose in the OPEL framework. This regularization promotes transportation between instances that exhibit homogeneity in the embedding space while simultaneously favoring temporal proximity. This approach recognizes that semantically similar actions (those with similar embeddings) that occur close in time are more likely to represent true correspondences than semantically similar actions that are temporally distant.

This regularization addresses a fundamental challenge in video analysis where similar-looking actions may occur multiple times throughout a sequence, but only certain instances represent true correspondences between videos. By incorporating both semantic similarity (through embedding space homogeneity) and temporal information, the method can make more informed decisions about which frame correspondences are most meaningful for procedure learning.

Procedure Learning

Why OPEL?

Kullback-Leibler (KL) Divergence

The second regularization term employs KL-divergence with an exponentially decaying prior to achieve smooth alignment while maintaining conformity to both optimality and temporal priors. This sophisticated approach balances multiple competing objectives in the alignment process: maintaining optimal transport solutions, respecting temporal relationships, and ensuring smooth transitions between aligned segments.

The exponential decay component of this regularization reflects the intuitive notion that temporal relationships become less reliable as the distance between frames increases. This mathematical formulation captures the idea that while nearby frames in time are likely to have strong correspondence relationships, distant frames should have exponentially decreasing influence on alignment decisions. The KL-divergence framework ensures that the learned alignments remain probabilistically consistent while incorporating these temporal priors.

OPEL

Mathematical Background

Handling background and redundant frames

To effectively manage background and redundant frames, the paper has integrated an additional '*virtual frame*' within the transport matrix T. This serves as a placeholder for aligning any frame that do not match with the primary sequence, and allows OPEL to explicitly assign these non-contributing frames to the virtual frame. The augmented transport matrix, now denoted as:

$$\hat{\boldsymbol{T}} \in \mathbb{R}^{(N+1)\times(M+1)}$$

This includes an extra row and column to accommodate the virtual frame. Note, if the likelihood of a frame aligning with any salient frame falls below a predefined threshold, ζ, we assign that frame to the virtual frame.

OPEL

Mathematical Background

Inverse Difference Moment Regularization

The IDM regularization term combines these components through: $M(\hat{\boldsymbol{T}}) = \phi\, M_t(\hat{\boldsymbol{T}}) + (1-\phi)\, M_o(\hat{\boldsymbol{T}})$

Temporal Coherence Term (M_t):

$$M_t(\hat{\boldsymbol{T}}) = \sum_{i=1}^{N+1} \sum_{j=1}^{M+1} \frac{t_{ij}}{\left(\frac{i}{N+1} - \frac{j}{M+1}\right)^2 + 1}$$

Penalizes transport between temporally distant frames

Optimality Coherence Term (M_o):

$$M_o(\hat{\boldsymbol{T}}) = \sum_{i=1}^{N+1} \sum_{j=1}^{M+1} \frac{t_{ij}}{\frac{1}{2} d_m + 1}, \quad \text{where } d_m = \left(\frac{i - i_o}{N+1}\right)^2 + \left(\frac{j - j_o}{M+1}\right)^2$$

Rewards transport aligned with OT-derived optimal positions (i_o,j_o).

To encourage optimal alignment, $M(\hat{T})$ of the learned $\hat{T}$ should be maximized.

OPEL

Mathematical Background

K-L Divergence

The KL Divergence(D_{KL}) is a type of statistical distance that measures how a model probability distribution Q diverges or is different from a true distribution P.

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}$$

Key properties:

- Non-symmetric: $D_{KL}(P||Q) \neq D_{KL}(Q||P)$
- Non-negative: $D_{KL} \geqslant 0$
- Zero only when $P = Q$

OPEL

Mathematical Background

- Regularization using K-L Divergence

  Exponentially Decaying Prior (Q): $\boldsymbol{Q}(i,j) = \phi\, \boldsymbol{Q}_t(i,j) + (1-\phi)\, \boldsymbol{Q}_o(i,j)$

  where ϕ serves as a dynamic weight, initially set to 1.0, and progressively reduced to 0.5 during training. This enables primarily temporal coherence at first and at later stages, balanced semantic-temporal alignment.

  Temporal Prior (Q_t):

$$\boldsymbol{Q}_t(i,j) = \frac{1}{2b} e^{-\frac{|d_t(i,j)|}{b}}, \quad \text{where } d_t(i,j) = \frac{|i/N - j/M|}{\sqrt{1/N^2 + 1/M^2}}$$

  A 2-dimensional Laplace distribution enforces exponentially decaying alignment likelihood with temporal distance. The scale parameter b controls temporal strictness (empirically set to 2.0-3.0). It promotes alignment of one sequence with elements in proximal temporal positions of the other sequence, thereby preserving the overall temporal structure and maintaining consistency in action order.
  The temporal proximity between frames is modelled using normalized temporal positions $d_t(i,j)$, where N,M are sequence lengths. This normalization handles videos of varying durations.

OPEL

Mathematical Background

- Regularization using K-L Divergence

Optimality Prior (Q_o):

$$\boldsymbol{Q}_o(i,j) = \frac{1}{2b} e^{-\frac{|d_o(i,j)|}{b}}, \quad \text{where } d_o(i,j) = \frac{|i/N - i_o/N| + |j/M - j_o/M|}{2\sqrt{1/N^2 + 1/M^2}}$$

Modelled as a 2-D Laplace Distribution, this prior represents the average distance from (i,j) to the frame locations (i,j_o) and (i_o,j) that correspond to the optimal alignment as indicated by the transport matrix, and b is a scale parameter.

This prior leverages the transport matrix T, which provides a preliminary indication of alignment between two video sequences. The point representing the most likely alignment, according to T, has the highest likelihood, while the assignment probability decays along any perpendicular direction from this centre.

OPEL

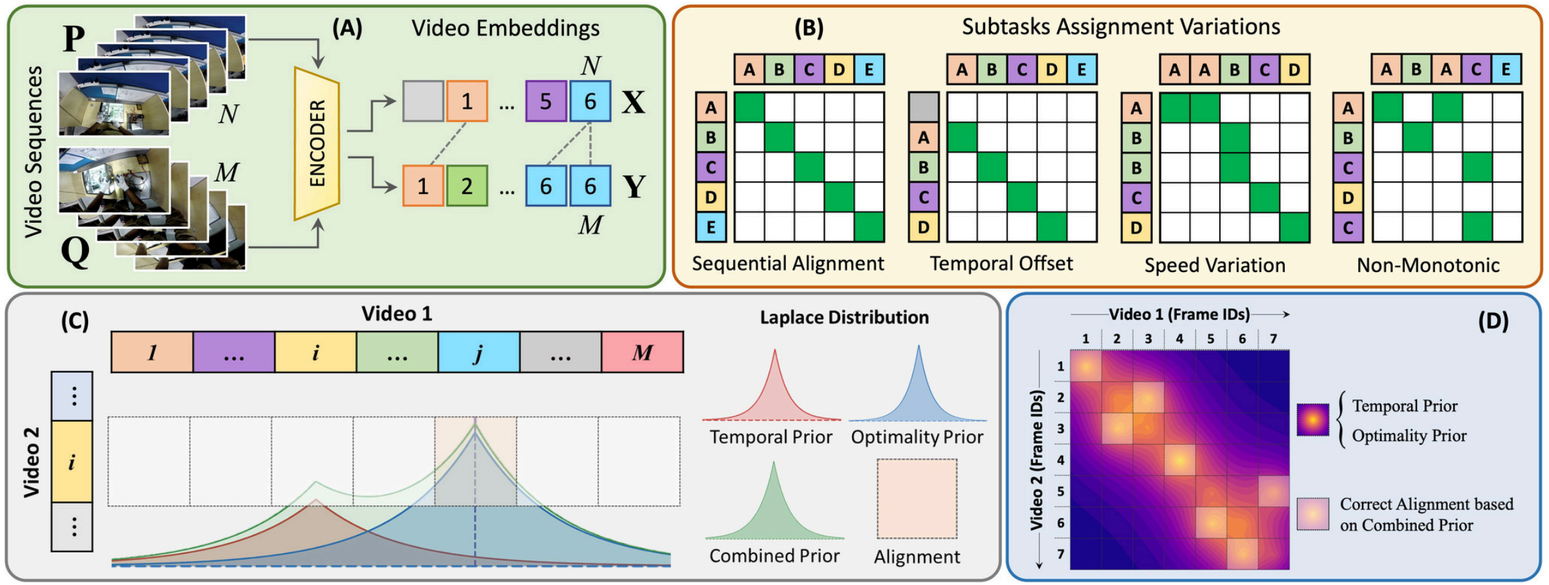
Mathematical Background

Comparison between the two divergences

<u>Inverse Difference Moment Regularisation</u>

- Speed Variation Tolerance: Handles videos where the same actions occur at different speeds.
- Repeated Action Alignment: Correctly aligns multiple instances of similar actions (e.g., multiple "stirring" steps)
- Local Smoothness: Ensures neighboring frames align with nearby temporal positions.

Figure is on the right
(A) The encoder generates frame-wise embeddings from videos, facilitating subsequent OT calculations.
(B) Pair-wise scenarios captured through the assignment matrix- from strictly synchronized actions to temporal shifts and differing action speeds, to non-monotonicity.
(C) 1-D depiction of alignment of a single frame (i-th) of Video 2 with its best match frame (j-th) of Video 1, based on the proposed priors.
(D) 2-D representation of the optimal alignment of frame sequences.



OPEL

Mathematical Background

- Comparison between the two divergences

  <u>K-L Divergence using priors</u>

  - Long-Range Noise Suppression: Prevents spurious alignments between temporally distant frames

  - Background Frame Filtering: Works with virtual frames to ignore irrelevant content.
    The augmented prior $\hat{Q}$ includes probability distributions for the virtual frame. For background frames, the prior assigns a higher probability to virtual frame alignment. KL-divergence pulls the transport matrix toward this structure.

  - Structural Smoothness: Maintains overall flow of procedural sequences. Avoids fragmented filtering that could disrupt procedural flow. Uses Probabilistic Filtering, maintaining smooth transitions between salient and non-salient content, keeping global coherence while filtering irrelevant content.

OPEL

Mathematical Background

Training the model

Assume, the inputs are two sequences of video frames: $\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_N]$ and $\boldsymbol{Q} = [\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_M]$ Passing these through a deep encoder network to obtain their respective embeddings, $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N]$ and $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_M]$ .Considering the elements of X and Y as independent samples, their probability measures can be written as, $f = \sum_{i=1}^{N} \alpha_i \delta_{x_i}$ and $g = \sum_{j=1}^{M} \beta_j \delta_{y_j}$, where δ_x denotes the Dirac mass at x, and α and β are the weights for the distributions f and g, respectively. Initially we set $\alpha_i = \frac{1}{N}$ and $\beta_j = \frac{1}{M}$ for all i,j. We create a matrix T where each element T_{ij} tells how much of frame x_i should be aligned with y_j given two conditions: Each row sums to $\alpha_i = 1/N$, Each column sums to $\beta_j = 1/M$. This set of constraints is called the transportation polytope:

$$U(\boldsymbol{\alpha}, \boldsymbol{\beta}) := \{\boldsymbol{T} \in \mathbb{R}_+^{N \times M} : \boldsymbol{T}\mathbf{1}_M = \boldsymbol{\alpha}, \boldsymbol{T}^\top \mathbf{1}_N = \boldsymbol{\beta}\}$$

Here, t_{ij} can be interpreted to be proportional to the probability that x_i will be aligned to y_j. This clearly depicts the need for the two conditions: initially, we're distributing $\alpha_i = \frac{1}{N}$ probability to M video frames for each of the i^{th} frame of the first video embeddings.

OPEL

Mathematical Background

Training the model

D, distance matrix of the size N×M, formed by computing the Euclidean Distances between the embedding vectors, $d(\boldsymbol{x}_i, \boldsymbol{y}_j) = \|\boldsymbol{x}_i - \boldsymbol{y}_j\|$. The cost of transporting mass from f to g with a transport plan T is quantified by the *Frobenius inner product* ⟨T , D⟩. Thus, the Wasserstein distance raised to the power p is: $W_p^p(f,g) = l_W(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{D}) = \min_{\boldsymbol{T} \in U(\boldsymbol{\alpha}, \boldsymbol{\beta})} \langle \boldsymbol{T}, \boldsymbol{D} \rangle$; p = 1 is taken in this paper. This is very hectic to train, so *Cuturi* introduced entropy regularization to solve this, using *Sinkhorn distance*,

$$l_\lambda^S(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{D}) = \langle \boldsymbol{T}_\lambda, \boldsymbol{D} \rangle \quad \text{s.t. } \boldsymbol{T}_\lambda = \arg \min_{\boldsymbol{T} \in U(\boldsymbol{\alpha}, \boldsymbol{\beta})} \langle \boldsymbol{T}, \boldsymbol{D} \rangle - \frac{1}{\lambda} h(\boldsymbol{T})$$

$$h(\boldsymbol{T}) = -\sum_{i=1}^{N} \sum_{j=1}^{M} t_{ij} \log t_{ij}$$

where h(T) denotes the entropy of T and λ is the regularization parameter. Solving this equation we get the Optimal Trasport Map T

OPEL

Mathematical Background

Incorporating the Regularizations

After adding the two regularizations as stated before, the equation becomes:

$$U_{\xi_1,\xi_2}(\boldsymbol{\alpha},\boldsymbol{\beta}) = \left\{\hat{\boldsymbol{T}} \in \mathbb{R}_+^{N+1\times M+1} \mid \hat{\boldsymbol{T}}\mathbf{1}_{M+1} = \boldsymbol{\alpha}, \hat{\boldsymbol{T}}^\top \mathbf{1}_{N+1} = \boldsymbol{\beta}, M(\hat{\boldsymbol{T}}) \geq \xi_1, \mathrm{KL}(\hat{\boldsymbol{T}} \parallel \hat{\boldsymbol{Q}}) \leq \xi_2\right\}$$

where $\hat{Q}$ is the augmented prior with virtual frame. The new Wasserstein distance between X and Y is:

$$l^R_{\xi_1,\xi_2}(\boldsymbol{X},\boldsymbol{Y}) = \min_{\hat{\boldsymbol{T}}\in U_{\xi_1,\xi_2}(\boldsymbol{\alpha},\boldsymbol{\beta})} \langle \hat{\boldsymbol{T}}, \boldsymbol{D}\rangle$$

Note: To encourage optimal alignment, Q(T) of the learned T should be maximized. So the divergence term has to be minimized:

$$\mathrm{KL}(\hat{\boldsymbol{T}} \parallel \hat{\boldsymbol{Q}}) = \sum_{i=1}^{N+1}\sum_{j=1}^{M+1} t_{ij}\log\frac{t_{ij}}{q_{ij}}$$

The above optimised equation can be solved by considering its dual equation (see below) by incorporating two Lagrange multipliers $\lambda_1 > 0$ and $\lambda_2 > 0$:

$$l^R_{\lambda_1,\lambda_2}(\boldsymbol{X},\boldsymbol{Y}) := \langle \hat{\boldsymbol{T}}_{\lambda_1,\lambda_2}, \boldsymbol{D}\rangle, \text{ s.t. } \hat{\boldsymbol{T}}_{\lambda_1,\lambda_2} = \arg\min_{\hat{\boldsymbol{T}}\in U(\boldsymbol{\alpha},\boldsymbol{\beta})} \langle \hat{\boldsymbol{T}}_{\lambda_1,\lambda_2}, \boldsymbol{D}\rangle - \lambda_1 M(\hat{\boldsymbol{T}}) + \lambda_2 \mathrm{KL}(\hat{\boldsymbol{T}} \parallel \hat{\boldsymbol{Q}}).$$

OPEL

Mathematical Background

Contrastive Regularization

Incorporating temporal priors into the video alignment processes often leads to trivial solutions. Without contrastive regularization, OPEL would learn useless embeddings where:

- All frames look identical in the embedding space
- The model can't tell the difference between "mixing" and "pouring"
- Everything gets mapped to the same point

So *Contrastive-Inverse Difference Moment (C-IDM)* loss is needed to further regularize the training.

Intra-Video Contrastive Loss

$$I(\boldsymbol{X}) = \sum_{i=1}^{N+1} \sum_{j=1}^{M+1} (1 - \mathcal{N}(i,j))\, \gamma(i,j) \max(0, \lambda_3 - d(i,j)) + \mathcal{N}(i,j) \frac{d(i,j)}{\gamma(i,j)}$$

where $\gamma(i,j) = (i-j)^2 + 1$, $d(i,j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$, $\mathcal{N}(i,j)$ is a neighborhood function defined as: $\mathcal{N}(i,j) = 1$, if $|i-j| \leq \delta$ and 0 otherwise, δ is a predefined window size, λ_3 is a margin parameter

OPEL

Mathematical Background

Intra-Video Contrastive Loss

Push Term for Distant Frames:

$$(1 - \mathcal{N}(i,j))\,\gamma(i,j)\max(0, \lambda_3 - d(i,j))$$

This term activates when |i-j| > δ (frames are temporally distant):

- Applies a hinge loss with margin λ_3
- Only penalizes when d(i,j) < λ_3 (embedding distance in Euclidian Space is smaller than the Margin Parameter, but the i^{th} frame and the j^{th} frame are already distant so its effect has to be penalised; otherwise, when the embedding distance is very high, previous methods are enough to remove the effects of the distant frames to each other)
- Penalty is proportional to temporal distance via γ(i,j)
- Creates a repulsive force that increases with temporal distance which helps to remove it's effect.

OPEL

Mathematical Background

Intra-Video Contrastive Loss

Pull Term for Distant Frames: $\mathcal{N}(i,j)\frac{d(i,j)}{\gamma(i,j)}$

This term activates when |i-j| <= δ (frames are temporally close):

- Directly penalizes embedding distance d(i,j) (if frames are temporally close then their embeddings should not be distant
- Inverse weighting by γ(i,j) gives higher importance to frames that are closer in time
- Creates an attractive force that decreases with temporal distance

Example:-
Frame 1: Cracking egg (time: 0:10)
Frame 2: Mixing batter (time: 0:15)
Frame 3: Pouring into pan (time: 0:45)

Without intra-video loss:

- All three frames might get the same embedding
- Model can't distinguish between different cooking steps

With intra-video loss:

- Nearby frames (1 & 2) are pulled together → similar embeddings
- Distant frames (1 & 3) are pushed apart → different embeddings
- Result: Model learns "cracking" and "mixing" are related, but "pouring" is different

OPEL

Mathematical Background

Inter-Video Contrastive Loss

$x_{\text{best}}(i) = \arg\max_j \hat{\boldsymbol{T}}_{\lambda_1,\lambda_2}$ and $x_{\text{worst}}(i) = \arg\min_j \hat{\boldsymbol{T}}_{\lambda_1,\lambda_2}$. Likewise, $y_{\text{best}}(j) = \arg\max_i \hat{\boldsymbol{T}}_{\lambda_1,\lambda_2}$ and $y_{\text{worst}}(j) = \arg\min_i \hat{\boldsymbol{T}}_{\lambda_1,\lambda_2}$ are calculated. Then, the best distance is computed as the average of squared differences between matched pairs, scaled by a temperature factor: best_distance $= \frac{1}{\text{temperature}} \cdot \left(\frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{y}_{x_{\text{best}}(i)}\|^2 + \frac{1}{M} \sum_{j=1}^{M} \|\boldsymbol{y}_j - \boldsymbol{x}_{y_{\text{best}}(j)}\|^2 \right)$. Similarly, the worst distance is: worst_distance $= \frac{1}{\text{temperature}} \cdot \left(\frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{y}_{x_{\text{worst}}(i)}\|^2 + \frac{1}{M} \sum_{j=1}^{M} \|\boldsymbol{y}_j - \boldsymbol{x}_{y_{\text{worst}}(j)}\|^2 \right)$. Finally, the inter-sequence loss is computed using the cross-entropy over the best and worst distances:

$$\text{loss\_inter} = F_{\text{cross\_entropy}} \left(\begin{bmatrix} \text{best\_distance} \\ \text{worst\_distance} \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right)$$

The cross-entropy loss treats this as a binary classification problem:

- Target for best_distance: 0 (perfect similarity)
- Target for worst_distance: 1 (maximum dissimilarity)

OPEL

Mathematical Background

Inter-Video Contrastive Loss

Ultimately, loss_inter=−log(1−best_distance)−log(worst_distance)). It implements the core contrastive learning principle:

- Pull together: Frames that should correspond (best matches)
- Push apart: Frames that shouldn't correspond (worst matches)

Ideally, we want each frame embedding x_i to align highly with its best match from Y. So, the best distance should be as close to 0 as possible; at the same time, we maximise its distance from the unmatched frame embeddings, and the same holds true for y_j's as we are doing bidirectional mapping (i.e. from Video 1 → Video 2 & Video 2 → Video 1). As a result, our proposed inter-video loss promotes learning *Disentangled Representations.*

Note: *Disentangled Representations* refers to the process of learning a representation of data where the individual factors of variation in the data are captured by separate, distinct elements of the representation. In simpler terms, it's about breaking down complex data into its underlying factors in a way that each factor is represented independently of the others.

OPEL

Mathematical Background

Overall Loss Function for OPEL

Overall loss for OPEL combines the regularized OT loss with the contrastive regularization terms:

$$L_{\text{OPEL}}(\boldsymbol{X}, \boldsymbol{Y}) = c_1 * l^R_{\lambda_1,\lambda_2}(\boldsymbol{X}, \boldsymbol{Y}) + c_2 * (I(\boldsymbol{X}) + I(\boldsymbol{Y})) + c_3 * \text{loss\_inter}$$

Ideally, we want each frame embedding x_i to align highly with its best match from Y. So, the best distance should be as close to 0 as possible; at the same time, we maximise its distance from the unmatched frame embeddings, and the same holds true for y_j's as we are doing bidirectional mapping (i.e. from Video 1 → Video 2 & Video 2 → Video 1). As a result, our proposed inter-video loss promotes learning *Disentangled Representations.*

Note: *Disentangled Representations* refers to the process of learning a representation of data where the individual factors of variation in the data are captured by separate, distinct elements of the representation. In simpler terms, it's about breaking down complex data into its underlying factors in a way that each factor is represented independently of the others.

OPEL

Mathematical Background

Clustering and Key-Step Ordering

After minimizing the final OPEL loss function, we obtain a learned optimal transport matrix along with discriminative frame embeddings that capture both semantic content and temporal structure. Each element t_{ij} indicates the strength of correspondence between frame i from one video and frame j from another video.

Now the goal is to *localize the key-steps required for PL.* This problem is framed as multi-label graph-cut segmentation. The node set V of the graph includes *k terminal nodes* representing *the key-steps* and *non-terminal nodes* corresponding to the number of frames, which are derived from the 128-dimentional embeddings produced by the ResNet-50 (Pretrained on ImageNet) embedder network. The embedder is trained using pairs of training videos. Within these videos, the authors randomly selected frames and optimized the proposed L_{OPEL} until convergence.

Upon constructing the graph, α-Expansion was applied to identify the minimum cost cut, utilising the results to assign frames to k labels.

OPEL

Mathematical Background

Clustering and Key-Step Ordering

Deducing the sequential order and cannonical ranking of the key-steps

To deduce the sequential order of key-steps, firstly the normalized time for each frame in a video is computed. Subsequently, the temporal instant for each cluster is determined by calculating the average normalized time for frames allocated to that cluster. Clusters are then sequenced in ascending order of their average time, thus outlining the sequence of key-steps of a video.

Upon establishing the key-step order for all videos associated with the same task, the paper generates a ranked list based on the frequency at which subjects adhere to a specific sequence. The most commonly observed order is placed at the top of this list. This methodological approach allows us to discern various sequential orders of key-steps of a task. The frequency-based ranking establishes standard operating procedures by identifying the most commonly followed sequence of steps.

OPEL

Results

Comparison with other PL Methods

Dataset - EgoProceL (1st Person View)

The 1st-person EgoProceL benchmark contains 62 hours of egocentric video recordings from 130 subjects engaged in 16 tasks.

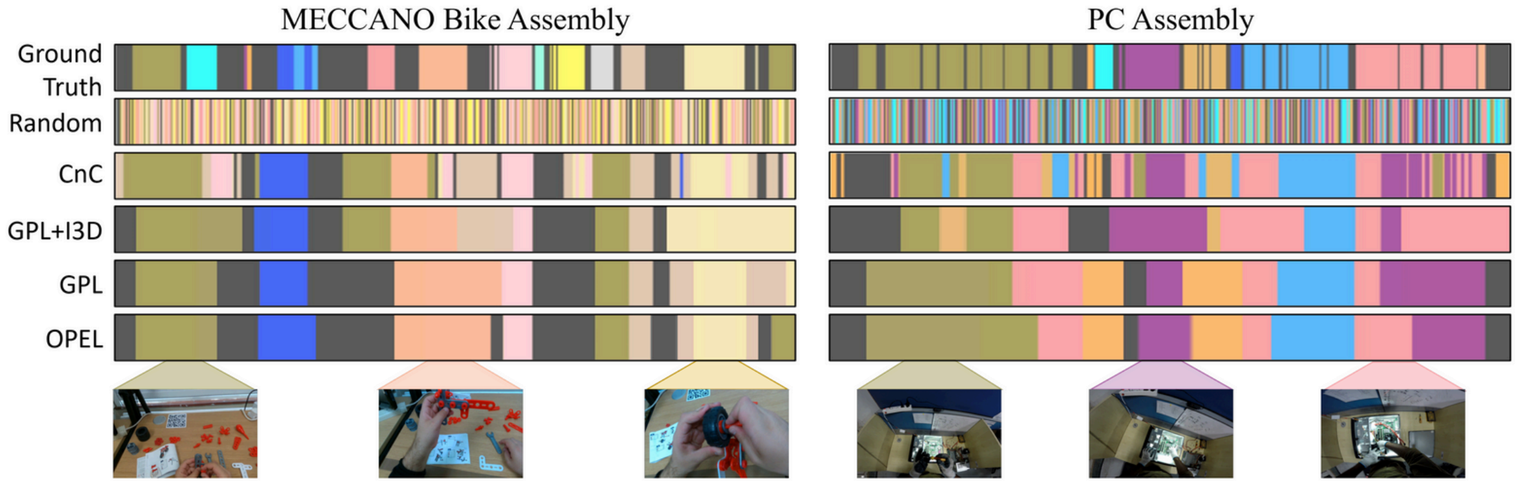
	EgoProceL											
	CMU-MMAC [17]		EGTEA-GAZE+[52]		MECCANO[53]		EPIC-Tents[54]		PC Assembly		PC Disassembly	
	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU
Random	15.7	5.9	15.3	4.6	13.4	5.3	14.1	6.5	15.1	7.2	15.3	7.1
Uniform	18.4	6.1	20.1	6.6	16.2	6.7	16.2	7.9	17.4	8.9	18.1	9.1
CnC [1]	22.7	11.1	21.7	9.5	18.1	7.8	17.2	8.3	25.1	12.8	27.0	14.8
GPL-2D [2]	21.8	11.7	23.6	14.3	18.0	8.4	17.4	8.5	24.0	12.6	27.4	15.9
UG-I3D [2]	28.4	15.6	25.3	14.7	18.3	8.0	16.8	8.2	22.0	11.7	24.2	13.8
GPL-w BG [2]	30.2	16.7	23.6	14.9	20.6	9.8	18.3	8.5	27.6	14.4	26.9	15.0
GPL-w/o BG [2]	31.7	17.9	27.1	**16.0**	20.7	10.0	19.8	9.1	27.5	15.2	26.7	15.2
OPEL *(Ours)*	**36.5**	**18.8**	**29.5**	13.2	**39.2**	**20.2**	**20.7**	**10.6**	**33.7**	**17.9**	**32.2**	**16.9**

OPEL

Results

Comparison with other PL Methods

Dataset - EgoProceL (1st Person View)



Qualitative results from MECCANO and PC Assembly tasks. Each sub-task is color-coded to represent different key-steps, while gray areas signify background elements. Notably, OPEL's performance surpasses that of the SOTA networks, attributed to its ability to handle unmatched frames through the integration of a virtual frame, thus enhancing alignment accuracy.

OPEL

Results

Comparison with other PL Methods

Dataset - CrossTask and ProceL (3rd Person View)

CrossTask features 213 hours of video footage spanning 18 primary tasks, totaling 2763 videos. ProceL includes 47.3 hours of video from 12 varied tasks, comprising 720 videos.

	ProceL [3]			CrossTask [11]		
	P	R	F1	P	R	F1
Uniform	12.4	9.4	10.3	8.7	9.8	9.0
Alayrc *et al.* [34]	12.3	3.7	5.5	6.8	3.4	4.5
Kukleva *et al.* [32]	11.7	30.2	16.4	9.8	35.9	15.3
Elhamifar *et al.* [3]	9.5	26.7	14.0	10.1	**41.6**	16.3
Fried *et al.* [37]	-	-	-	-	28.8	-
Shen *et al.* [47]	16.5	31.8	21.1	15.2	35.5	21.0
CnC [1]	20.7	22.6	21.6	22.8	22.5	22.6
GPL-2D [2]	21.7	23.8	22.7	24.1	23.6	23.8
UG-I3D [2]	21.3	23.0	22.1	23.4	23.0	23.2
GPL [2]	22.4	24.5	23.4	24.9	24.1	24.5
STEPS [16]	23.5	26.7	24.9	26.2	25.8	25.9
OPEL *(Ours)*	**33.6**	**36.3**	**34.9**	**35.6**	34.8	**35.1**

OPEL

Results

- Comparison with models with multimodal input

  Dataset - EgoProceL (1st Person View)

  The 1st-person EgoProceL benchmark contains 62 hours of egocentric video recordings from 130 subjects engaged in 16 tasks.

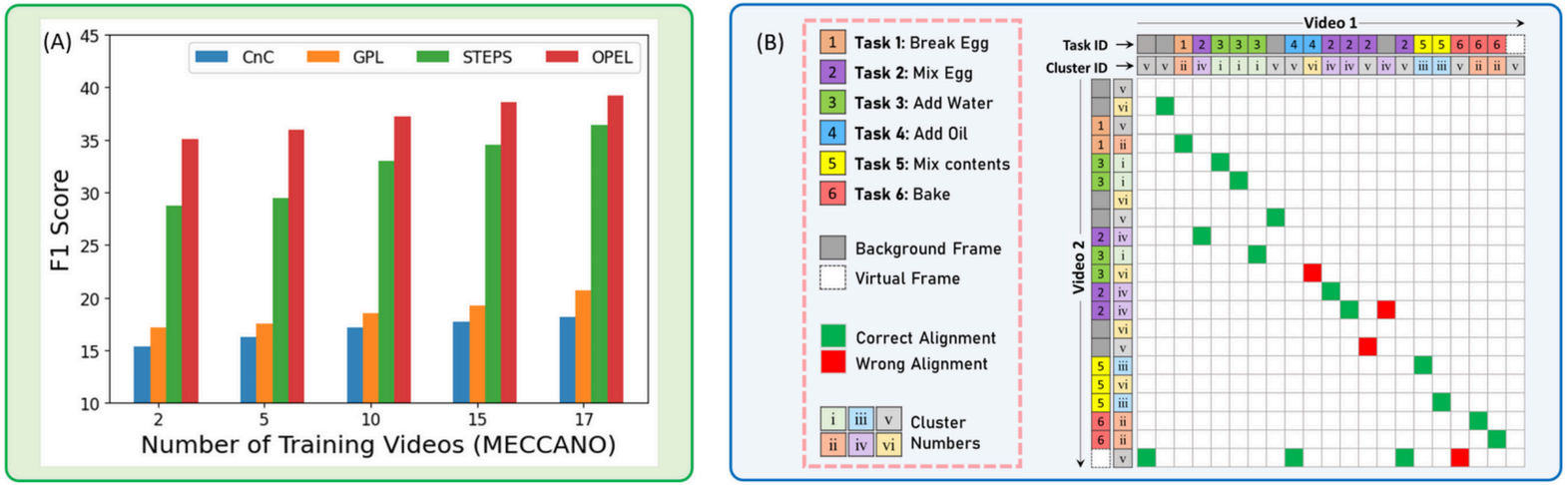
	CMU-MMAC		EGTEA-GAZE+		MECCANO		EPIC-Tents		ProceL		CrossTask	
	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU
STEPS [16]	28.3	11.4	**30.8**	12.4	36.4	18.0	**42.2**	**21.4**	24.9	15.4	25.9	14.6
OPEL	**36.5**	**18.8**	29.5	**13.2**	**39.2**	**20.2**	20.7	10.6	**34.9**	**21.3**	**35.1**	**21.5**

OPEL

Results

Quantitative Comparison

Dataset - EgoProceL (1st Person View)



(A) Impact of training data quantity on encoder training.
(B) Example alignment of two videos with corresponding key-step clusters from the Brownie task

Thank you!