# DMGT: Composable Relational Learning from Heterogeneous Developmental Snapshots

Anonymous authors

September 26, 2026

**Abstract**

Single-cell developmental studies provide snapshots of evolving cell populations rather than longitudinal measurements of individual cells, so developmental relationships are only partially observable. Existing computational approaches typically model one source of developmental information, such as transcriptional similarity, temporal progression, or lineage relationships, but lack a unified representation learning formulation for integrating **heterogeneous relations** that vary across experimental settings.

We introduce DMGT (Developmental Multi-relational Graph Transformer), a **relation-composable representation learning framework** for heterogeneous developmental snapshots. DMGT models cells as nodes in developmental graphs where available biological relations, including **state similarity, temporal ordering, and lineage context**, are explicitly represented as distinct edge types and jointly integrated through a **relation-aware graph Transformer**. On lineage-resolved datasets, the resulting representations support three predictive objectives: developmental fate prediction, lineage-consistent cross-time correspondence, and auxiliary prediction of the next-observed same-clone molecular state. On barcode-free datasets, the same encoder is pre-trained with a masked-expression objective and evaluated through a frozen linear probe on later-time state annotation.

We evaluate DMGT across five single-cell RNA-seq datasets spanning lineage-resolved and barcode-free regimes. Temporal edges improve Macro-F1 for fate prediction and AUROC for cross-time correspondence, and they also improve later-time state annotation on the unbarcoded datasets; under a single regression objective they slightly increase future molecular-state error. The effect of adding a relation is not fixed: the Full configuration underperforms simpler graphs when trained on the regression loss alone but gives the best molecular-state prediction when trained jointly with the fate and correspondence losses. These results indicate a **relation-objective interaction** between relations and training objectives, and argue for **configuring edges and objectives together rather than uniformly aggregating every available relation**.

## 1 Introduction

Development unfolds as cells traverse molecular states and commit to distinct identities. Yet single-cell technologies observe this process through destructive population snapshots: each measurement destroys the cell, so no individual cell is tracked across time. Computational methods must therefore recover developmental organization from collections of cell populations sampled at discrete stages, where the relationships between observations are only partially known.

A central difficulty is that developmental evidence is heterogeneous and experiment-dependent. Transcriptional similarity reveals state organization, temporal sampling imposes directional ordering, and lineage tracing provides experimentally measured clonal constraints. Different experiments expose different subsets of these relations: some datasets include lineage barcodes, others record only molecular profiles at multiple time points, and none universally provides all sources. Existing methods typically optimize for one such signal at a time. Pseudotime ordering reconstructs progression from transcriptional similarity (Trapnell et al., 2014), RNA velocity estimates directional dynamics from splicing kinetics (La Manno et al., 2018; Bergen et al., 2020), optimal transport maps populations across stages (Schiebinger et al., 2019; Klein et al., 2025), and lineage-aware methods connect clonal history to future outcomes (Weinreb et al., 2020; Biddy et al., 2018; Lange et al., 2022; Wang et al., 2022). What is missing is a representation learning formulation that can jointly integrate whichever biological relations are available in a given experiment, rather than committing to a fixed inference objective or a single transition assumption.

From a graph learning perspective, this means the set of relation types is not fixed across datasets. Standard heterogeneous graph methods assume a predefined and consistent set of edge types (Kipf & Welling, 2017; Schlichtkrull et al., 2018; Hu et al., 2020), whereas developmental datasets require the relation set itself to vary. The key insight is to treat each biological relation as a configurable edge type in a shared graph, so that the same representation learner operates on whatever subset of relations an experiment provides. We call this property *relation-composable*: the encoder architecture is unchanged



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

across relation configurations, and only the graph construction selects which edge types are present.

We introduce DMGT (Developmental Multi-relational Graph Transformer), a relation-composable framework for learning predictive cellular representations from developmental snapshots. DMGT represents cells as nodes in a graph whose edge types correspond to available biological relations: within-stage transcriptional similarity, directed temporal adjacency between stages, and, when lineage barcodes are present, same-clone context. A relation-aware graph Transformer aggregates messages across these edge types with relation-specific attention biases, producing contextualized cell representations. On lineage-resolved datasets, these representations are trained through three complementary objectives: fate prediction, lineage-consistent cross-time correspondence, and auxiliary prediction of the next-observed same-clone molecular state. On datasets without lineage information, the same encoder is pre-trained with a masked-expression objective and evaluated through a frozen linear probe on later-time state annotation.

Across five single-cell RNA-seq datasets spanning both regimes, we find that the utility of developmental relations depends on the training objectives. Temporal edges yield gains in fate prediction (Macro-F1 +0.113 and +0.037), cross-time correspondence (AUROC +0.021 and +0.055), and later-time state annotation on the unbarcoded datasets, while they slightly increase future molecular-state error when the encoder is trained on that regression objective alone. The dependence is more pronounced for lineage context: under future-only training the Full configuration underperforms StateOnly by 0.034 and 0.095 in MSE, but under joint three-objective training it attains the best MSE overall, 0.956 and 0.815. Aggregating every available relation is therefore neither uniformly helpful nor uniformly harmful; relations and objectives interact, and the two should be configured together, using validation data.

Our contributions are:

1. We formulate developmental representation learning as a relation-composable graph learning problem, in which heterogeneous biological relations are integrated as configurable edge types under a shared encoder.

2. We introduce DMGT, a relation-aware graph Transformer that learns predictive representations from configuration-specific developmental graphs and supports both lineage-resolved multi-task training and barcode-free self-supervised pre-training.

3. We evaluate DMGT across five datasets and two information regimes. We show that the utility of developmental relations is conditional rather than fixed: edge types and training objectives interact on the molecular-state task, and aggregating all relations helps only under joint multi-task training. The results argue for configuring relations and objectives together, on validation data, rather than uniformly aggregating edges or choosing them per task alone.

## 2 Related Work

### 2.1 Developmental dynamics from single-cell snapshots

Single-cell developmental studies typically observe heterogeneous populations across discrete stages rather than longitudinally tracking individual cells. Existing computational approaches have therefore focused on reconstructing latent developmental dynamics from snapshot measurements. Pseudotemporal ordering methods infer continuous developmental progressions from transcriptional similarity (Trapnell et al., 2014), while RNA velocity and its dynamical extensions estimate directional state transitions using transcriptional kinetics (La Manno et al., 2018; Bergen et al., 2020). Optimal transport based approaches further model developmental transitions as probabilistic mappings between populations, enabling inference of temporal and spatial cell-state transitions (Schiebinger et al., 2019; Klein et al., 2025).

While these methods have advanced trajectory reconstruction, they primarily focus on estimating transition structures or developmental ordering under specific assumptions. They generally do not address how multiple sources of developmental evidence, such as molecular similarity, temporal ordering, and lineage context, can be jointly represented within a unified predictive framework. DMGT differs by learning cellular representations through composable developmental relations rather than explicitly inferring a single transition trajectory.

### 2.2 Lineage-aware developmental modeling

Lineage tracing provides direct experimental measurements of cellular history and has revealed that transcriptionally similar cells may exhibit different future outcomes. Large-scale lineage tracing studies have combined clonal information with single-cell transcriptomics to connect cellular states with future fates (Weinreb et al., 2020; Biddy et al., 2018), while developmental barcoding approaches have enabled lineage recording at larger scales (Kalhor et al., 2018).

Recent computational methods incorporate lineage information for fate prediction and developmental mapping. CellRank integrates molecular states with directional transition models to estimate cell fate probabilities (Lange et al., 2022), and CoSpar combines transcriptomic dynamics with lineage tracing information to identify early fate biases (Wang et al., 2022). Unlike CellRank and CoSpar, which integrate lineage information into a transition-matrix estimation framework, DMGT incorporates lineage as an explicit edge type in a graph Transformer and jointly optimizes representation learning across fate, correspondence, and future-state objectives. DMGT also supports datasets where lineage information is absent by removing clone-specific edges and using a masked-expression pre-training objective.

### 2.3 Relational representation learning

Graph neural networks provide a general framework for learning from structured relational data. Graph convolutional networks learn node representations through neighborhood aggregation (Kipf and Welling, 2017), while relational graph

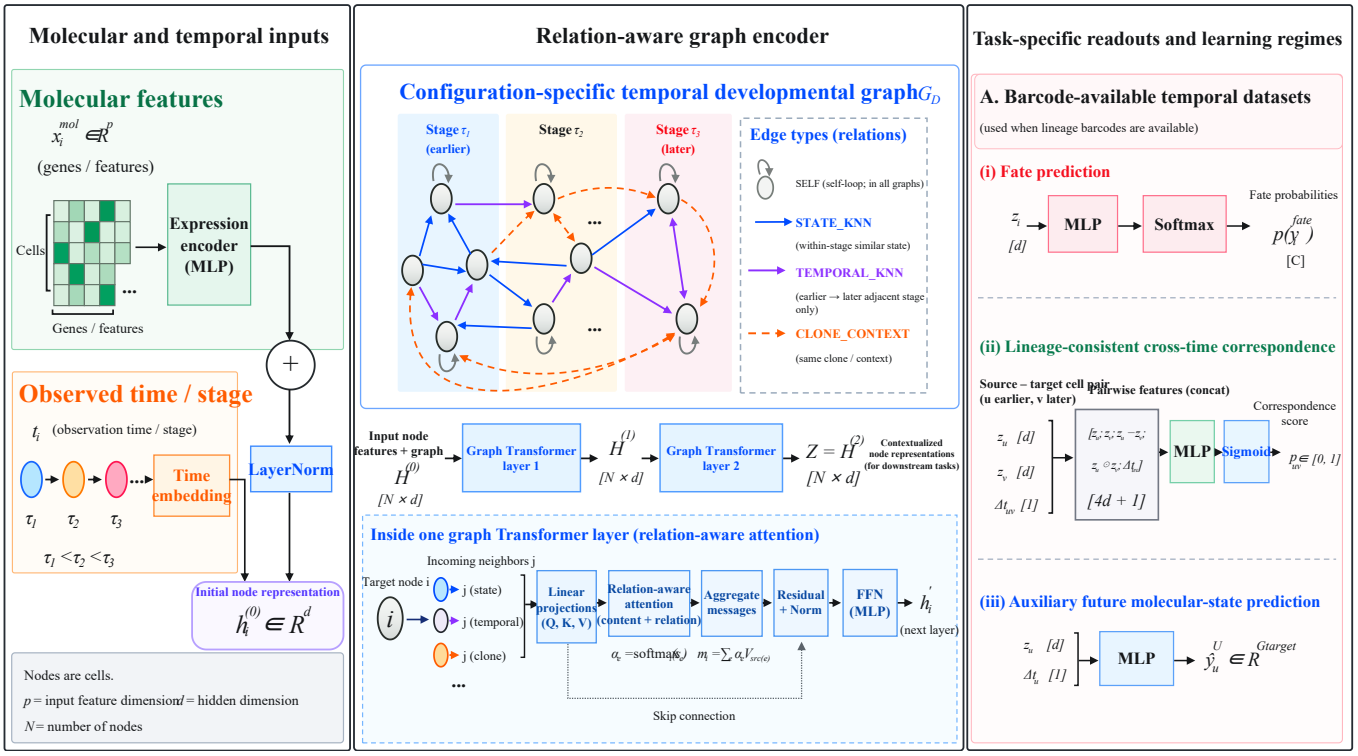



**Figure 1.** Overview of DMGT. DMGT learns predictive cellular representations from heterogeneous developmental snapshots by composing available biological relations. Cells are embedded from molecular states and observation times and organized into relation-configurable developmental graphs containing state, temporal, and lineage relations. A relation-aware graph Transformer produces contextualized representations. On lineage-resolved datasets, representations are evaluated through fate prediction, lineage-consistent correspondence, and future-state prediction; on barcode-free datasets, the encoder is evaluated via a frozen linear probe on later-time state annotation.

neural networks extend graph learning to multiple edge types (Schlichtkrull et al., 2018). Recent graph Transformer models further enable relation-aware attention over heterogeneous graphs (Hu et al., 2020; Ying et al., 2021). These architectures have also been applied to single-cell data for tasks such as cell clustering and multi-omics integration.

However, existing relational learning approaches typically assume a fixed set of relation types across all datasets. Developmental single-cell datasets introduce a different challenge: available relations vary across experimental designs, with some datasets providing lineage information while others contain only molecular and temporal observations. DMGT addresses this challenge by constructing relation-configurable developmental graphs, where biological relations are selected according to available evidence and evaluated through multiple predictive objectives.

## 3 Method

### 3.1 Overview

DMGT learns predictive representations from heterogeneous developmental snapshots by explicitly modeling the biological relations available in each experiment. Unlike conventional graph learning approaches that assume a fixed relational structure, DMGT constructs configuration-specific developmental graphs, where molecular similarity, developmental ordering, and lineage context are represented as distinct relations.

Given a developmental snapshot dataset

$$\mathcal{D} = \{(x_i, t_i, m_i)\}_{i=1}^{N},$$

where $x_i$ denotes molecular measurements, $t_i$ denotes the observed developmental stage, and $m_i$ represents optional metadata such as lineage identity, DMGT represents cells as nodes in a relational graph

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R}),$$

where each edge is associated with a biological relation type. The available relation set is dataset-dependent:

$$\mathcal{R} \subseteq \{\text{SELF}, \text{STATE}, \text{TEMPORAL}, \text{CLONE}\}.$$

This formulation enables a shared representation learner across lineage-resolved and barcode-free developmental datasets.

### 3.2 Molecular-temporal representation and graph construction

Each molecular profile is projected into a latent representation:

$$h_i^{mol} = f_\theta(x_i),$$

where $f_\theta$ is an expression encoder. Since developmental time provides directional information beyond transcriptional similarity, DMGT introduces a learnable temporal embedding:

$$e_i^{time} = E(t_i).$$

The initial node representation is:

$$h_i^{(0)} = \mathrm{LayerNorm}(h_i^{mol} + e_i^{time}).$$

Based on these representations, DMGT constructs developmental graphs according to available biological evidence. The STATE relation captures within-stage molecular similarity, while TEMPORAL introduces directed connections between adjacent developmental stages:

$$t_a < t_b, \quad (v_a, v_b) \in \mathcal{E}_{TEMPORAL}.$$

When lineage information is available, CLONE connects cells sharing the same experimentally observed clone identity. Importantly, clone relations encode shared lineage context rather than deterministic parent–child ancestry.

### 3.3 Relation-aware graph Transformer encoder

DMGT adopts a relation-aware graph Transformer to integrate information from heterogeneous developmental relations. For each edge type, the model learns a relation-specific attention bias so that messages from STATE, TEMPORAL, and CLONE neighbors are weighted differently.

For node $i$ and neighbor $j$ connected by relation $r_{ij}$, the attention score is computed as:

$$s_{ij} = \frac{(W_Q h_i)^T (W_K h_j)}{\sqrt{d}} + b_{r_{ij}},$$

where $b_{r_{ij}}$ is a learnable relation-specific bias. The aggregated message is:

$$m_i = \sum_{j \in \mathcal{N}(i)} \alpha_{ij} W_V h_j,$$

where

$$\alpha_{ij} = \mathrm{softmax}_j(s_{ij}).$$

Each layer applies residual aggregation and feed-forward transformation:

$$h_i^{(l+1)} = \mathrm{FFN}(\mathrm{Norm}(h_i^{(l)} + m_i)).$$

After $L$ layers, the contextualized representation is:

$$z_i = h_i^{(L)}.$$

### 3.4 Developmental prediction objectives

The following three objectives are defined for lineage-resolved datasets where clone identity is available. The masked-expression pre-training objective used on barcode-free datasets is defined in Section 3.5.

**Fate prediction.** For lineage-resolved datasets with experimentally derived fate labels, DMGT predicts developmental identity:

$$\hat{y}_i^{fate} = f_{fate}(z_i).$$

**Lineage-consistent correspondence.** To evaluate developmental continuity, DMGT predicts whether two observations correspond to the same clone across developmental stages. A positive pair is defined as:

$$y_{ij}^{corr} = 1 \iff c_i = c_j \wedge t_j > t_i.$$

The correspondence score is:

$$p_{ij} = f_{corr}(z_i, z_j, \Delta t_{ij}).$$

This objective evaluates whether representations preserve experimentally observed cross-time developmental relationships.

**Future molecular-state prediction.** As an auxiliary predictive probe, DMGT predicts the mean molecular profile of cells sharing the same clone identity at the next observed developmental stage. For clone $c$, the target is defined as:

$$y_i^{future} = \frac{1}{|C_{c,t^+}|} \sum_{j \in C_{c,t^+}} x_j,$$

where $t^+$ denotes the next observed developmental time point. The prediction is:

$$\hat{y}_i^{future} = f_{future}(z_i).$$

### 3.5 Masked-expression pre-training

For datasets that provide no lineage barcodes, DMGT is pre-trained by reconstructing withheld expression values. For each cell $i$, a set $\mathcal{M}_i$ of gene coordinates is sampled without replacement at a fixed masking ratio $\rho$, and the corresponding entries are replaced in the encoder input by a learned mask indicator. The encoder maps the partially observed profile, together with its temporal embedding and whatever graph edges are constructed, to a latent representation; a reconstruction head $f_{\mathrm{rec}}$ then predicts the masked coordinates:

$$\tilde{x}_{ig} = f_{\mathrm{rec}}(z_i), \qquad g \in \mathcal{M}_i.$$

The pre-training objective is the mean squared error over the masked entries only:

$$\mathcal{L}_U = \frac{1}{\sum_i |\mathcal{M}_i|} \sum_{i=1}^{N} \sum_{g \in \mathcal{M}_i} (\tilde{x}_{ig} - x_{ig})^2,$$

evaluated on the log-normalized expression used throughout graph construction. Because $\mathcal{L}_U$ is defined per node, it

applies unchanged whether the graph contains STATE and TEMPORAL edges or no edges at all.

### 3.6 Optimization

The overall training objective combines the predictive losses with the masked-expression loss:

$$\mathcal{L} = \lambda_F \mathcal{L}_F + \lambda_C \mathcal{L}_C + \lambda_{\text{fut}} \mathcal{L}_{\text{fut}} + \lambda_U \mathcal{L}_U,$$

where $\mathcal{L}_F$, $\mathcal{L}_C$, and $\mathcal{L}_{\text{fut}}$ denote the fate, correspondence, and future molecular-state losses. Each term is included only when the labels or observations it requires are present: barcode-free datasets reduce the objective to $\mathcal{L}_U$, whereas lineage-resolved datasets activate the predictive losses and may additionally retain $\mathcal{L}_U$ as a regularizer. Just as the edge set is configured per experiment, the set of optimized objectives is configured per experiment; the two are chosen independently at construction time and jointly determine the learned representations.

## 4 Experimental Setup and Results

We evaluate DMGT on five single-cell RNA-seq developmental datasets covering two complementary biological regimes: lineage-resolved systems with experimentally observed clone information (GSE140802, GSE99915) and barcode-free systems without lineage measurements (GSE67310, GSE94883, GSE132188). These datasets enable us to examine how different developmental relations contribute to predictive representation learning under varying information availability.

For lineage-resolved datasets, we evaluate three tasks: fate prediction, lineage-consistent cross-time correspondence, and auxiliary future molecular-state prediction. Correspondence positives are defined as same-clone pairs where the target is observed at a later time point, indicating shared clone context rather than direct parent–child ancestry. Future-state prediction uses the mean molecular profile of same-clone cells at the next observed stage as the target. For barcode-free datasets, we pre-train the encoder with a masked-expression objective and evaluate it through a frozen logistic-regression probe on later-time state annotation. Across all experiments, we compare DMGT with expression-based predictors, relational graph baselines (R-GCN, PyG-HGT), and transition-based approaches (WOT, moscot, CellRank, CoSpar) under their corresponding information regimes.

### 4.1 Datasets and evaluation protocol

Table 1 summarizes the five datasets. GSE140802 (49,116 cells, 3 stages: D2/D4/D6) and GSE99915 (18,803 cells, 6 stages: D6–D28) provide lineage barcodes and are used for the lineage-resolved evaluation. GSE67310 (405 cells, 5 stages: D0/D2/D5/D20/D22), GSE94883 (1,964 cells, 5 stages: D2–D6), and GSE132188 (3,696 cells, 4 stages: E12.5–E15.5) do not include lineage information and are used for the barcode-free evaluation.

**Data splits.** For lineage-resolved datasets, we use a clone-disjoint split (70% train / 10% validation / 20% test) so that no clone appears in more than one partition. For barcode-free datasets, we use a stage-holdout split where earlier stages are used for training and the latest stage is held out for testing, evaluating whether the learned representations generalize to unseen developmental times.

**Relation configurations.** We evaluate three relation configurations. StateOnly includes SELF and STATE edges (within-stage molecular similarity). Strict adds TEMPORAL edges (directed adjacency between adjacent stages) while excluding clone-context edges. Full additionally includes CLONE edges and is therefore an *oracle* setting: it uses lineage information at inference time, which may not be available in deployment. All configurations share the same encoder architecture.

**Evaluation metrics.** Fate prediction is evaluated with Macro-F1 and balanced accuracy. Cross-time correspondence is evaluated with AUROC, average precision (AP), and mean reciprocal rank (MRR) over source–target candidate sets. Future molecular-state prediction is evaluated with mean squared error (MSE). Barcode-free state annotation is evaluated with Macro-F1 and balanced accuracy using a frozen logistic-regression readout on the pre-trained representations. All results are averaged over 3 random seeds.

### 4.2 Fate prediction on lineage-resolved data

Table 2 compares fate prediction performance across methods. DMGT–Strict achieves Macro-F1 of 0.636 and 0.663 on GSE140802 and GSE99915, compared with 0.540 and 0.584 for the closest generic graph baseline PyG–HGT–Strict. The balanced accuracy of DMGT–Strict is 0.592 and 0.650, versus 0.560 and 0.585 for PyG–HGT–Strict.

Relative to DMGT–StateOnly, adding temporal edges (Strict) improves Macro-F1 by +0.113 on GSE140802 and +0.037 on GSE99915, while balanced accuracy changes by only +0.006 and +0.019. Because balanced accuracy is the macro-average of per-class recall, this pattern indicates that temporal edges mainly reduce false-positive fate assignments and thereby raise precision; the fraction of cells recovered for each class, including minority fates, changes little.

The Full (oracle) configuration reaches Macro-F1 0.659 and 0.674 and balanced accuracy 0.641 and 0.700. These results use clone-context edges at inference time and should be interpreted as an upper bound when lineage information is available, rather than the expected performance in a deployment setting without lineage access.

### 4.3 Lineage-consistent cross-time correspondence

Table 3 evaluates pair discrimination and source-conditioned retrieval. DMGT–Strict records AUROC/AP/MRR of 0.768/0.298/0.335 on GSE140802 and 0.810/0.324/0.338 on GSE99915. Relative to PyG–HGT–Strict, the AUROC differences are +0.042 and +0.109, and the MRR differences are

**Table 1.** Overview of developmental datasets used for evaluating DMGT.

| Dataset | Cells | Genes | Barcode | # stages | Sampling labels |
|---|---|---|---|---|---|
| GSE67310 | 405 | 13,202 | No | 5 | D0, D2, D5, D20, D22 |
| GSE132188 | 3,696 | 27,998 | No | 4 | E12.5, E13.5, E14.5, E15.5 |
| GSE94883 | 1,964 | 19,009 | No | 5 | D2, D3, D4, D5, D6 |
| GSE140802 | 49,116 | 25,289 | Yes | 3 | D2, D4, D6 |
| GSE99915 | 18,803 | 28,001 | Yes | 6 | D6, D9, D12, D15, D21, D28 |

**Table 2.** Fate prediction on lineage-resolved developmental datasets under different relational configurations.

| Method / configuration | GSE140802 | | | GSE99915 | | |
|---|---|---|---|---|---|---|
| | Macro-F1 ↑ | BalAcc ↑ | $n_s$ | Macro-F1 ↑ | BalAcc ↑ | $n_s$ |
| Expression LR | 0.485 | 0.527 | 3 | 0.524 | 0.529 | 3 |
| Expression MLP | 0.512 | 0.534 | 3 | 0.513 | 0.515 | 3 |
| R–GCN–Strict† | 0.525 | 0.514 | 3 | 0.556 | 0.542 | 3 |
| PyG–HGT–Strict† | 0.540 | 0.560 | 3 | 0.584 | 0.585 | 3 |
| DMGT–StateOnly | 0.523 | 0.586 | 3 | 0.626 | 0.631 | 3 |
| DMGT–Strict | 0.636 | 0.592 | 3 | 0.663 | 0.650 | 3 |
| DMGT–Full (oracle)‡ | **0.659** | **0.641** | 3 | **0.674** | **0.700** | 3 |

† Generic graph baselines evaluated under the Strict relation configuration (STATE + TEMPORAL, no clone edges).
‡ Full uses additional clone-context edges at inference time and is not an equal-information control.

+0.135 and +0.110, showing improvements in both discrimination and ranking.

Relative to DMGT–StateOnly, adding temporal edges (Strict) improves AUROC by +0.021 and +0.055. The larger performance gap is between DMGT and the generic graph baselines rather than between DMGT configurations. Since R–GCN and PyG–HGT already perform relation-aware aggregation on the same Strict edges, this gap is attributable to global Transformer-style attention and the multi-task objectives rather than to relation awareness itself.

The Full (oracle) configuration reaches AUROC 0.871 and 0.890 and MRR 0.389 and 0.385. Because clone-context edges directly connect same-clone cells and clone agreement defines the positive target, this setting is near-circular and is reported only as an upper bound.

The unsupervised transition and transport estimators (WOT, moscot, CellRank–RTK, CoSpar–StateOnly) form a separate comparison stratum: they do not use pair-label supervision. CoSpar–StateOnly achieves the highest AUROC among this group (0.657 and 0.672), but remains below DMGT–Strict (0.768 and 0.810). The gap reflects both architectural and supervision differences; a fully matched comparison would require training the transport estimators with the same pair-label objective.

### 4.4 State annotation without lineage supervision

Table 4 evaluates whether the learned representations remain useful when clone-context edges are unavailable. The encoder is pre-trained with a masked-expression objective on training stages, frozen, and evaluated with a logistic-regression probe on the held-out later stage. We report Macro-F1 and balanced accuracy, where $K$ denotes the number of annotated cell-state classes used as prediction targets.

In the frozen-representation block, DMGT–ST + LR achieves the highest scores on all three datasets: Macro-F1 0.556, 0.561, and 0.546 for GSE67310, GSE94883, and GSE132188, with balanced accuracy 0.594, 0.596, and 0.578. The corresponding PCA + LR values are 0.322, 0.342, and 0.366, while PyG–HGT–ST + LR gives 0.490, 0.498, and 0.473. DMGT–ST + LR outperforms PyG–HGT–ST + LR by +0.066, +0.063, and +0.073 Macro-F1 across the three datasets.

Comparing DMGT–ST with DMGT–S (which excludes temporal edges), the Macro-F1 differences are +0.051, +0.020, and +0.081, and the balanced-accuracy differences are +0.049, +0.024, and +0.035. The temporal-edge benefit is positive on all three datasets but varies in magnitude, indicating that the strength of cross-stage context is dataset-dependent.

The top block of Table 4 reports directly supervised predictors (Majority, Expression LR, Expression MLP) that train end-to-end on the annotation task. These use a different training regime than the frozen-representation block and are not directly comparable; they are included as reference points for the difficulty of the annotation task.

### 4.5 Auxiliary future molecular-state prediction

Table 5 evaluates whether the learned representations can predict the mean molecular profile of same-clone cells at the next observed stage. On GSE140802, DMGT–StateOnly and DMGT–Strict report MSE 1.050 and 1.057, compared with 1.080 for source-state Ridge and 1.090 for source-state MLP. On GSE99915, the corresponding values are 0.850 and 0.877, compared with 0.920 and 0.898. DMGT–StateOnly slightly outperforms both source-state baselines on both datasets.

**Table 3.** Cross-time correspondence prediction using lineage-aware developmental representations.

| Method / configuration | GSE140802 | | | GSE99915 | | |
|---|---|---|---|---|---|---|
| | AUROC ↑ | AP ↑ | MRR ↑ | AUROC ↑ | AP ↑ | MRR ↑ |
| *Pair-supervised, no clone-context* | | | | | | |
| Pairwise LR | 0.629 | 0.116 | 0.156 | 0.636 | 0.152 | 0.126 |
| Pairwise MLP | 0.641 | 0.155 | 0.117 | 0.652 | 0.153 | 0.133 |
| R–GCN–Strict† | 0.653 | 0.180 | 0.198 | 0.692 | 0.185 | 0.190 |
| PyG–HGT–Strict† | 0.726 | 0.242 | 0.200 | 0.701 | 0.231 | 0.228 |
| DMGT–StateOnly | 0.747 | 0.285 | 0.245 | 0.755 | 0.310 | 0.282 |
| DMGT–Strict | **0.768** | **0.298** | **0.335** | **0.810** | **0.324** | **0.338** |
| *Unsupervised transition / transport estimators* | | | | | | |
| WOT | 0.559 | 0.060 | 0.072 | 0.541 | 0.135 | 0.109 |
| moscot | 0.610 | 0.114 | 0.131 | 0.582 | 0.128 | 0.129 |
| CellRank–RTK | 0.581 | 0.150 | 0.158 | 0.611 | 0.147 | 0.154 |
| CoSpar–StateOnly | 0.657 | 0.171 | 0.151 | 0.672 | 0.199 | 0.216 |
| DMGT–Full (oracle)‡ | **0.871** | **0.408** | **0.389** | **0.890** | **0.468** | **0.385** |

† Generic graph baselines evaluated under the Strict relation configuration.
‡ Full uses clone-context edges at inference time; since clone agreement defines the positive target, this is an oracle setting and not an equal-information comparison.

**Table 4.** Later-time state annotation from frozen representations without lineage supervision.

| Method / representation | GSE67310 ($K$=5) | | GSE94883 ($K$=5) | | GSE132188 ($K$=4) | |
|---|---|---|---|---|---|---|
| | Macro-F1 ↑ | BalAcc ↑ | Macro-F1 ↑ | BalAcc ↑ | Macro-F1 ↑ | BalAcc ↑ |
| *Directly supervised annotation predictors* | | | | | | |
| Majority | 0.082 | 0.200 | 0.097 | 0.250 | 0.068 | 0.167 |
| Expression LR | 0.417 | 0.433 | 0.379 | 0.392 | 0.360 | 0.399 |
| Expression MLP | 0.444 | 0.391 | 0.443 | 0.461 | 0.363 | 0.362 |
| *Frozen representations with common logistic-regression readout* | | | | | | |
| PCA + LR | 0.322 | 0.401 | 0.342 | 0.357 | 0.366 | 0.372 |
| Expression-only encoder + LR† | 0.453 | 0.476 | 0.400 | 0.454 | 0.361 | 0.431 |
| R–GCN–ST + LR† | 0.432 | 0.493 | 0.468 | 0.504 | 0.388 | 0.456 |
| PyG–HGT–ST + LR† | 0.490 | 0.478 | 0.498 | 0.477 | 0.473 | 0.485 |
| DMGT–S + LR | 0.505 | 0.545 | 0.541 | 0.572 | 0.465 | 0.543 |
| DMGT–ST + LR | **0.556** | **0.594** | **0.561** | **0.596** | **0.546** | **0.578** |

$K$ denotes the number of annotated cell-state classes used as prediction targets.
† Expression-only and generic graph frozen-encoder controls, evaluated with the same logistic-regression readout.

Adding temporal edges (STRICT) raises MSE by 0.007 on GSE140802 and 0.027 on GSE99915 relative to STATEONLY when the encoder is trained on the future-state loss alone. Under this single-objective regime, and in contrast to fate prediction and correspondence, temporal edges give no measurable benefit. The Full rows follow the same pattern under future-only training (1.084 and 0.945), yet the Full configuration trained jointly with all three objectives attains the lowest MSE in the table (0.956 and 0.815). We examine this dependence between relations and objectives in Section 4.6.

### 4.6 Interaction between relations and training objectives

The analyses above vary edge types and training objectives one at a time. The two Full rows of Table 5 hold the edge configuration fixed and vary only the objectives optimized during training, which isolates how the objectives alter the value of the available relations on the future-state task.

Under single-objective training, adding every available relation does not help. Full–Future-only attains MSE 1.084 and 0.945, compared with 1.050 and 0.850 for StateOnly, a change of +0.034 and +0.095. The identical edge configuration trained jointly with the fate and correspondence losses (JOINT3) attains 0.956 and 0.815, a reduction of 0.128 and 0.130 relative to Future-only and the lowest MSE in Table 5. The strongest configuration on this task is therefore neither the sparsest graph nor the fullest graph on its own, but the fullest graph combined with multi-task training.

TEMPORAL and CLONE edges extend each node's neighborhood across stages and across clonally related cells. With the future-state loss as the only signal, messages from that enlarged neighborhood average over descendants that have already diverged along several fates; the clone-mean target supplies no gradient that separates them, and the predictions move toward the stage-wise population mean. The fate and correspondence losses provide that missing signal: the fate loss penalizes confusion between identities, while the correspondence loss contrasts same-clone pairs against unrelated cross-time pairs, anchoring the aggregated messages to clone- and fate-specific structure.

**Table 5.** Predicting next-observed same-clone molecular states from developmental representations.

| Predictor / training configuration | GSE140802 | | GSE99915 | |
|---|---|---|---|---|
| | MSE ↓ | $n_s$ | MSE ↓ | $n_s$ |
| Persistence (carry-forward) | 1.158 | 3 | 1.027 | 3 |
| Mean-future (train only) | 1.477 | 3 | 1.311 | 3 |
| Source-state Ridge | 1.080 | 3 | 0.920 | 3 |
| Source-state MLP | 1.090 | 3 | 0.898 | 3 |
| DMGT–StateOnly | 1.050 | 3 | 0.850 | 3 |
| DMGT–Strict | 1.057 | 3 | 0.877 | 3 |
| DMGT–Full / JOINT3 (oracle)‡ | **0.956** | 3 | **0.815** | 3 |
| DMGT–Full / Future-only (oracle)‡ | 1.084 | 3 | 0.945 | 3 |

The target is the mean molecular profile of same-clone cells at the next observed stage, not a later measurement of the same physical cell.
‡ Full configurations use clone-context edges at inference time (oracle setting).

The marginal value of a relation is consequently conditional on the objectives it is trained with. Edges that are redundant, or marginally harmful, under a single-task regime become useful under joint multi-task training, and the two should be configured together on the validation set rather than composed independently. A fully crossed factorial study would estimate the corresponding interaction terms directly and offers a natural setting in which to learn relation–objective gating rather than configure it by hand.

## 5 Conclusion

We studied predictive representation learning from heterogeneous developmental snapshots, where biological relations are incomplete and vary across experimental settings. We introduced DMGT, a relation-composable framework that represents cells through configuration-specific relational graphs and integrates available molecular-state, temporal, and lineage information when provided.

Our results indicate that the utility of developmental relations is conditional rather than fixed. Temporal edges improve fate prediction and cross-time correspondence and support later-time state annotation, while they add no measurable benefit to molecular-state prediction under a single regression objective. More tellingly, the Full configuration underperforms simpler graphs when trained on the regression loss alone but gives the best molecular-state prediction when trained jointly with the fate and correspondence losses. We read this as an interaction between dense cross-time relations and discriminative training objectives: the former broaden the set of developmental hypotheses available to each cell, while the latter prevent message passing from averaging those hypotheses away.

DMGT offers a modular framework for composing biological relations in developmental representation learning. In practical terms, relations and objectives should be configured together, using validation data, rather than aggregated without distinction. A key direction for future work is to learn this co-selection, gating edge types and objectives instead of configuring them manually, together with fully crossed factorial experiments that quantify the interaction, and extensions to spatial transcriptomics and multi-omics data.

## AI use statement

Generative AI tools were used during the development of this manuscript to assist with research brainstorming, organization of research ideas, and discussion of potential methodological considerations. These tools were used as auxiliary support for improving the clarity and structure of the manuscript.

All experimental design decisions, dataset processing procedures, implementation choices, statistical analyses, numerical results, figures, and scientific conclusions were independently determined, executed, and verified by the authors based on the actual experiments and experimental records. The manuscript was written and reviewed by the authors, who take full responsibility for the accuracy and integrity of the reported work.

## References


C. Trapnell, D. Cacchiarelli, J. Grimsby, P. Pokharel, S. Li, M. Morse, N. J. Lennon, K. J. Livak, T. S. Mikkelsen, and J. L. Rinn. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology*, 32:381–386, 2014.

G. La Manno, R. Soldatov, A. Zeisel, E. Braun, H. Hochgerner, V. Petukhov, K. Lidschreiber, M. E. Kastriti, P. Lönnerberg, A. Furlan, et al. RNA velocity of single cells. *Nature*, 560:494–497, 2018.

V. Bergen, M. Lange, S. Peidli, F. A. Wolf, and F. J. Theis. Generalizing RNA velocity to transient cell states through dynamical modeling. *Nature Biotechnology*, 38:1408–1415, 2020.

G. Schiebinger, J. Shu, M. Tabaka, B. Cleary, V. Subramanian, A. Solomon, J. Gould, S. Liu, S. Lin, P. Berube, et al. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943.e22, 2019.

D. Klein, G. Palla, M. Lange, M. Klein, Z. Piran, M. Gander, L. Meng-Papaxanthos, M. Sterr, L. Saber, C. Jing, et al. Mapping cells through time and space with moscot. *Nature*, 638:1065–1075, 2025.

C. Weinreb, A. Rodriguez-Fraticelli, F. D. Camargo, and A. M. Klein. Lineage tracing on transcriptional landscapes links state to fate during differentiation. *Science*, 367:eaaw3381, 2020.

B. A. Biddy, W. Kong, K. Kamimoto, C. Guo, S. E. Waye, T.-L. Sun, and S. A. Morris. Single-cell mapping of lineage and identity in direct reprogramming. *Nature*, 564:219–224, 2018.

R. Kalhor, K. Kalhor, L. Mejia, K. Leeper, A. Graveline, P. Mali, and G. M. Church. Developmental barcoding of whole mouse via homing CRISPR. *Science*, 361:eaat9804, 2018.

T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks. *International Conference on Learning Representations*, 2017.

P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018.

M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling. Modeling Relational Data with Graph Convolutional Networks. *European Semantic Web Conference*, pages 593–607, 2018.

Z. Hu, Y. Dong, K. Wang, and Y. Sun. Heterogeneous Graph Transformer. *Proceedings of The Web Conference*, pages 2704–2710, 2020.

C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu. Do Transformers Really Perform Bad for Graph Representation? *Advances in Neural Information Processing Systems*, 34:28877–28888, 2021.

A. Vaswani, N. Shazeer, N. Parmar, N. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and P. Polosukhin. Attention Is All You Need. *Advances in Neural Information Processing Systems*, 30, 2017.

R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15:1053–1058, 2018.

F. A. Wolf, P. Angerer, and F. J. Theis. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, 19:15, 2018.

I. Virshup, D. Bredikhin, L. Heumos, G. Palla, G. Sturm, A. Gayoso, and others. The scverse project provides a computational ecosystem for single-cell omics data analysis. *Nature Biotechnology*, 41:604–606, 2023.

I. Virshup, A. Rybakova, F. J. Theis, and others. AnnData: A Python package for handling annotated data matrices. *Journal of Open Source Software*, 6(61):3256, 2021.

B. Treutlein, Q. Q. Lee, J. Camp, M. Mall, W. Koh, S. A. Shariati, S. Sim, N. F. Neff, J. M. Skotheim, M. Wernig, and S. R. Quake. Dissecting direct reprogramming from fibroblast to neuron using single-cell RNA sequencing. *Nature*, 534:391–395, 2016.

A. Bastidas-Ponce, K. Scheibner, M. Tarquis-Medina, and others. Comprehensive single-cell mRNA profiling reveals a detailed roadmap for pancreatic endocrinogenesis. *Development*, 146:dev173849, 2019.

M. Lange, V. Bergen, M. Klein, M. Setty, B. Reuter, M. Bakhti, H. Lickert, F. J. Theis, S. Peidli, and others. CellRank for directed single-cell fate mapping. *Nature Methods*, 19:159–170, 2022.

S. Wang, M. J. Herriges, K. Hurley, D. N. Kotton, and A. M. Klein. CoSpar identifies early cell fate biases from single-cell transcriptomic and lineage information. *Nature Biotechnology*, 40:1066–1074, 2022.