mathematics

MDPI

Article

Sinkhorn-Optimized Capacity-Constrained Temporal Heterogeneous Graph Model for Startup–Investor Matching

Bosen Yang [1] and Chao He [2,*]

[1] School of Journalism, Chongqing University, Chongqing 400044, China; 20241641@stu.cqu.edu.cn
[2] Graduate School of Technology and Management, Kyung Hee University, Seoul 03722, Republic of Korea
* Correspondence: chaohe@khu.ac.kr

Abstract

Investment capacity couples startup–investor recommendations, but independent scoring and ranking ignores this coupling, repeatedly recommending a few investors. We formulate quarterly matching as capacity-constrained entropic optimal transport and propose CapTHG. A relation-specific heterogeneous attention encoder processes causal quarterly snapshots of a temporal heterogeneous graph, and a differentiable Sinkhorn layer converts its scores into a capacity-feasible transport plan. A dustbin handles upper-bound capacities, training is end-to-end with a plan likelihood loss, and an activity-based proxy from a recent window estimates capacity. We characterize the capacity-feasible set with the dustbin score as a retention threshold, derive the linear programming limit and error bound as regularization vanishes, and prove linear convergence and a gradient equal to the plan. In a quarterly rolling evaluation on real financing rounds from a public Crunchbase snapshot, CapTHG attains the highest round-level recall and normalized discounted cumulative gain among all baselines. For every encoder, capacity-constrained Sinkhorn ranking lowers capacity overload and raises investor coverage versus score ranking. Accuracy also improves only under joint training with the plan. Capacity and plan likelihood loss contribute most in the ablation study; results are stable across capacity definitions and candidate sets. Controlled numerical experiments agree with the theory, supporting transport-based capacity-constrained matching.

Keywords: startup–investor matching; capacity constraint; entropic optimal transport; Sinkhorn algorithm; temporal heterogeneous graph; recommendation; optimization and control; machine learning and data mining; network science

MSC: 49Q22; 68T07; 90C08

Check for updates

Academic Editor: Andrea Scozzari

Received: 22 September 2026
Revised: 16 October 2026
Accepted: 19 October 2026
Published: 22 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Venture capital channels scarce funds to young firms, and the way startups are paired with investors in each quarter shapes which of these firms obtain financing. Gompers et al. [1] surveyed venture capital firms and reported that the team, the industry, and the deal source drive investment decisions. Sørensen [2] modeled venture capital as a two-sided matching market in which investors and companies select each other. Both views imply that the recommendation for one startup–investor pair depends on the offers made to every other pair in the same quarter.

Graph learning provides most of the tools used for such pairings. Learned prediction models are used across many applied problems, for example, overexposed face super-

resolution with multi-scale exposure normalization and dual-branch aggregation [3], unsupervised deep learning-based damage identification under ambient excitations [4], forest fire risk rating forecasts based on XGBoost [5] and semi-supervised learning with pseudo labels [6], and physics-informed neural network prediction of the temperature field in high geothermal tunnels [7]. For recommendation, Wu et al. [8] surveyed graph neural networks in recommender systems and showed how message passing over user–item graphs improves ranking. Hu et al. [9] parameterized attention by node type and relation type in the heterogeneous graph Transformer, so that one encoder can read investors, startups, financing rounds and syndication ties within a single structure. For the optimization step, Cuturi [10] added an entropy term to optimal transport and solved the regularized problem by Sinkhorn matrix scaling at far lower cost than linear programming. Sarlin et al. [11] placed a differentiable Sinkhorn layer on top of graph neural network scores for feature matching and used dustbin rows and columns to absorb unmatched points. Christakopoulou et al. [12] addressed the economic side and built a recommendation model with capacity constraints for items of limited capacity.

An investor can take part in only a limited number of financing rounds in a quarter, and a financing round seats only a limited number of investors. Graph recommendation models score each startup–investor pair separately and then sort the scores. Under this procedure, one prominent investor can top thousands of lists at the same time. Investment capacity therefore has to enter the output of the model rather than a post hoc filter applied afterwards.

This paper formulates quarterly startup–investor matching as an entropic optimal transport problem whose marginals encode investment capacity. Capacity is measured by recent deal activity, namely the mean number of deals per quarter over the previous four quarters. This activity-based proxy reflects how many rounds an investor can enter in a quarter, since fund size and staffing are not observed. A relation-specific heterogeneous attention encoder reads causal quarterly snapshots of a temporal heterogeneous graph and produces matching scores. A differentiable Sinkhorn layer then converts these scores into a transport plan that respects capacity. The main model uses the equality-normalized form, and the dustbin-augmented form covers the general case in which capacity is only an upper bound. The model, CapTHG, is trained and evaluated on a public Crunchbase snapshot covering 2005–2015, which contains 48,921 financing rounds, 4763 candidate investors and 30,963 startups. In the quarterly rolling evaluation, CapTHG reaches a round-level Recall@20 of 0.3919, compared with 0.3216 for the strongest baseline R-GCN, a relative improvement of 21.9%. With the same encoder, replacing score ranking by capacity-constrained Sinkhorn ranking lowers the capacity overload ratio OR@10 from 0.5486 to 0.3246 and raises investor coverage Cov@10 from 0.3121 to 0.4204. The Sinkhorn ranking lowers overload and raises coverage for all six learned encoders compared. Recall@20 rises only for CapTHG, whose encoder is trained jointly with the plan.

Three research questions follow from this setting. RQ1 asks whether the per-quarter limits on the number of rounds per investor and the number of investors per round can be written as a differentiable matching layer with provable feasibility, limit and gradient properties. RQ2 asks whether a temporal heterogeneous graph encoder trained jointly with that layer recommends investors more accurately than comparable graph recommendation models on real financing rounds. RQ3 asks how the same layer changes over-capacity exposure and investor coverage, and how this change relates to accuracy.

The main contributions of this paper are as follows.

- We construct a capacity-constrained entropic optimal transport matching layer and prove four of its properties, namely the feasibility of the transport plan under the dustbin construction, the linear programming limit as $\varepsilon \to 0$ with an error bound, the

linear convergence of the Sinkhorn iteration, and the equality between the gradient of the layer and the transport plan.

- We design CapTHG, a temporal heterogeneous graph matching model whose relation-specific heterogeneous attention encoder runs on causal quarterly snapshots and is trained end to end with a plan likelihood loss, with investment capacity estimated from a four-quarter activity window.
- We conduct a quarterly rolling evaluation on the public Crunchbase snapshot, which shows that CapTHG improves accuracy over graph baselines and that the capacity-constrained Sinkhorn ranking lowers capacity overload and widens investor coverage, and we check robustness to the capacity definition, the candidate set, the mini-batch size and the hyperparameters.

2. Related Work

Startup–investor matching draws on two lines of research that have each advanced in recent years, namely representation learning on graphs and optimal transport. This section reviews heterogeneous and temporal graph learning for recommendation in Section 2.1, entropic optimal transport and Sinkhorn scaling in Section 2.2, and startup–investor matching together with capacity-constrained recommendation in Section 2.3.

2.1. Heterogeneous and Temporal Graph Learning for Recommendation

Graph neural networks have become the basic paradigm for modeling interactions in recommendation tasks. Wu et al. [8] gave a systematic review of graph neural networks in recommender systems, and Gao et al. [13] further organized the challenges, methods and future directions of this field. Among specific models, LightGCN, proposed by He et al. [14], removes nonlinear activation and feature transformation and keeps only neighbor aggregation and layer averaging.

Heterogeneous graph neural networks extend this idea to graphs in which several node types and relation types coexist. Schlichtkrull et al. [15] proposed R-GCN, which models relational data by assigning a separate weight matrix to each relation type. Wang et al. [16] proposed HAN, which applies node-level and semantic-level attention under each meta-path and has become a representative form of heterogeneous graph attention. Wang et al. [17] reviewed the methods, techniques, applications and resources of heterogeneous graph embedding. Wang et al. [18] built a self-supervised heterogeneous graph neural network with co-contrastive learning, and Lv et al. [19] revisited the benchmarks and implementations of this model family. In recent work published in Mathematics, Li et al. [20] introduced high-order semantics and node attention into a heterogeneous graph neural network for recommendation, and Du et al. [21] fused external features with internal attributes to predict enterprise bankruptcy.

Temporal graph models further bring the time of each interaction into the representation. Xu et al. [22] combined a functional time encoding with self-attention for inductive representation learning on temporal graphs, and their time encoding is the form adopted in this article. Wang et al. [23] addressed the same task with causal anonymous walks. The heterogeneous graph Transformer of Hu et al. [9] parameterizes attention by node type and relation type and introduces a relative time encoding. Fan et al. [24] proposed a heterogeneous temporal graph neural network, and Wang et al. [25] gave a simple and efficient version of it. Cao et al. [26] performed link prediction on temporal heterogeneous graphs based on the information lifecycle. In parallel, Cong et al. [27] questioned whether temporal networks need complicated architectures, Yu et al. [28] presented a new architecture and a unified library for dynamic graph learning, Huang et al. [29] established a benchmark for machine learning on temporal graphs, and Zheng et al. [30] surveyed

dynamic graph neural networks. Beyond recommendation, graph-based prediction has also been applied to diverse domains: Nguyen and Vo [31] integrated fuzzy-enhanced heterophilic graph representation learning to improve link prediction accuracy among heterogeneous nodes, Cui et al. [32] used directed hypergraphs to model high-order dependencies among skeleton joints for human pose forecasting, and Ye et al. [33] proposed an adaptive spatial-temporal-aware graph learning method for EEG-based emotion recognition that dynamically constructs electrode-level graph connections. Wang et al. [34] designed a merge-and-split graph convolutional network for multi-person skeleton-based interaction recognition, and Chen et al. [35] identified the topology of complex dynamical networks with time delays in finite time. These studies confirm the generality of graph neural network architectures across prediction tasks, while the recommendation setting imposes additional output-level constraints that the encoder alone cannot address.

The output stages of all the above models follow the same practice regardless of encoder design. Each model computes a matching score for every candidate pair, optimizes the ranking with the pairwise ranking loss of Rendle et al. [36], and takes the top candidates independently by score. Such an output treats each recommendation as an independent decision. It cannot express cross-item coupling, for example, that one investor can enter only a limited number of financing rounds in a quarter, and it does not write the capacity constraint into the model output.

2.2. Entropic Optimal Transport and Sinkhorn Scaling

Writing matching as a constrained allocation problem rests on matrix scaling and entropic optimal transport. Sinkhorn and Knopp [37] proved that a positive matrix can be brought to a doubly stochastic matrix by alternately scaling its rows and columns, which gave an existence basis for later iterative solvers. Cuturi [10] added an entropic regularization term to optimal transport and solves it with Sinkhorn matrix scaling. The computational cost is far below that of linear programming, so optimal transport could for the first time be called repeatedly on medium-scale problems. A monograph by Peyré and Cuturi [38] systematically covered entropic regularization, Sinkhorn iterations, dual potentials and convergence, and has become the standard reference of this line. In this framework, the row constraints fix the mass sent by each source point and the column constraints fix the mass received by each target point. The solution is a transport plan that satisfies both sets of marginal constraints.

The convergence rate and complexity of Sinkhorn iterations are supported by several results. Franklin and Lorenz [39] showed that matrix scaling converges linearly in the Hilbert projective metric with a contraction ratio bounded by the cross-ratio of the kernel matrix. Altschuler et al. [40] gave a complexity analysis of near-linear time approximation of optimal transport via Sinkhorn iterations. Thibault et al. [41] proposed the overrelaxed Sinkhorn–Knopp algorithm to further accelerate regularized optimal transport. Li et al. [42] solved the convex–concave saddle-point problem of group distributionally robust optimization with stochastic Mirror-Prox and reported that, at equal numbers of sampled records, stochastic mirror descent reaches a smaller mean duality gap than Mirror-Prox at mini-batch sizes from 2 to 128; Sinkhorn scaling, a mirror-descent-type scheme in the Kullback–Leibler geometry, belongs to the same broad family of first-order methods for constrained learning problems. Together, the results on Sinkhorn iterations show that, for a fixed regularization strength ε, a finite number of iterations brings the solution into a neighborhood of the marginal constraints. Sinkhorn iterations are therefore suitable for embedding in models that require repeated forward computation.

Relaxed marginal constraints are handled in two ways. Chizat et al. [43] proposed scaling algorithms for unbalanced optimal transport in which the marginal constraints

are relaxed into divergence penalties. Séjourné et al. [44] reviewed unbalanced optimal transport from theory to numerics. The second method augments the cost matrix with an extra row and column. Sarlin et al. [11] fed graph neural network scores into a differentiable Sinkhorn layer for feature matching and used a dustbin row and column to absorb unmatched points. This construction expresses the case in which the total mass may fall below the upper limit while keeping the equality-constrained form of the solution.

Differentiable Sinkhorn layers have since entered learning models for a range of matching and recommendation tasks. Yang et al. [45] used multimodal optimal transport for knowledge distillation in cross-domain recommendation. Chen et al. [46] proposed regularized discrete optimal transport for class-imbalanced classification. Chen et al. [47] aligned graphs via optimal transport with attribute interaction and self-training. Wang et al. [48] proposed graph-regularized survival learning with optimal transport. Montesuma et al. [49] summarized recent advances of optimal transport in machine learning. The marginal constraints in these works [45–48] are equalities or are relaxed as a whole by a divergence penalty. None of these works formulates a capacity constraint, such as the number of financing rounds that an investor can join per quarter, as a marginal constraint and handles its inequality form with a dustbin, and this line has not been brought into temporal heterogeneous graph recommendation.

2.3. Startup–Investor Matching and Capacity-Constrained Recommendation

Work on the startup investment side began with prediction. Żbikowski and Antosiuk [50] predicted business success from Crunchbase data, Ross et al. [51] used a machine learning model for startup selection and exit prediction, Kim et al. [52] also predicted startup success with machine learning, and Maarouf et al. [53] fused a large language model to predict startup success. Chen et al. [54] used panel data of artificial intelligence enterprises and a system GMM regression to show that a faster pace of across-industrial-chain investment raises the innovation performance of these enterprises, which suggests that the pace of investment also matters for the firms that receive it. The prediction methods above rank startups by success probability but do not answer which investor should act. Later studies moved the starting point to the investor side. Liang and Yuan [55] predicted investor funding behavior from Crunchbase social network features. Xu et al. [56] combined network diffusion with investors' domain preference to recommend investors for new startups. Hwang et al. [57] performed startup–investor matching with multimodal contrastive learning. The recommendation target has changed, yet the output of these models [55–57] still stops at a matching score for each pair.

Matching markets brought the preferences and quotas of both sides into the solution much earlier. Gale and Shapley [58] found a two-sided stable matching with the deferred acceptance algorithm, and Galichon and Salanié [59] characterized the social surplus in matching models and discussed its identification. Reciprocal recommendation carries this idea back into recommender systems. Palomares et al. [60] surveyed recommendation settings in which the preferences of both sides must be satisfied. Su et al. [61] optimized rankings for recommendation in matching markets, Tomita et al. [62] gave a fast and examination-agnostic reciprocal recommendation method, and Tomita and Yokoyama [63] studied fair reciprocal recommendation in matching markets. Most of these methods [61–63] work on static bipartite interactions and rarely describe relations that change as financing rounds proceed.

Capacity-constrained recommendation directly limits how often the objects on one side are recommended. Christakopoulou et al. [12] built a recommendation model with capacity constraints for settings in which the item side has limited capacity. Studies of bias provide evidence on the same issue. Chen et al. [64] reviewed bias and debiasing

in recommender systems, and Abdollahpouri et al. [65] evaluated popularity bias from a user-centered perspective. Unconstrained ranking repeatedly concentrates exposure on a few head objects. In startup investment, this means that a few head investors receive recommendations far beyond their investment capacity.

The three lines above each cover part of the problem. Startup investment studies compute matching scores without considering investment capacity, matching market studies respect the preferences and quotas of both sides on static graphs, and capacity-constrained recommendation controls exposure apart from the transport plan of Section 2.2 and the temporal heterogeneous graph encoder of Section 2.1. No prior work has formulated quarterly startup–investor matching as entropic optimal transport with capacity constraints, handles the inequality form of investment capacity with a dustbin, and trains the transport plan jointly with the graph encoder.

3. Problem Formulation

3.1. Temporal Heterogeneous Graph and Causal Quarterly Snapshots

The temporal heterogeneous graph in this study contains four node types, namely, investors $\mathcal{I}$, startups $\mathcal{C}$, sectors $\mathcal{S}$, and regions $\mathcal{R}$. The data span the period 2005–2015 and are divided into quarters. The graph comprises 4763 investors, 30,963 startups, 834 sectors, 905 regions, 48,921 financing rounds and 108,272 investment edges. The node set is given by Equation (1).

$$\mathcal{V} = \mathcal{I} \cup \mathcal{C} \cup \mathcal{S} \cup \mathcal{R} \tag{1}$$

The relation type set $\mathcal{T}$ contains eight directed relation types so that $|\mathcal{T}| = 8$. Its composition is given by Equation (2).

$$\mathcal{T} = \{\, \mathcal{I}\to\mathcal{C},\ \mathcal{C}\to\mathcal{I},\ \mathcal{C}\to\mathcal{S},\ \mathcal{S}\to\mathcal{C},\ \mathcal{C}\to\mathcal{R},\ \mathcal{R}\to\mathcal{C},\ \mathcal{I}\to\mathcal{R},\ \mathcal{R}\to\mathcal{I} \,\} \tag{2}$$

The relation types $\mathcal{I} \to \mathcal{C}$ and $\mathcal{C} \to \mathcal{I}$ are the two directions of the investment edge. Each investment edge e carries the quarterly time stamp $q(e)$ of its financing round. The remaining six relation types describe the sector and region membership of startups and investors. These static edges do not change from quarter to quarter. All investors in the same financing round share one time stamp, so syndication is observed in the same quarter.

The causal quarterly snapshot is written as $G_q = (\mathcal{V}, \mathcal{E}_q, \mathcal{T})$. Its edge set $\mathcal{E}_q$ is given by Equation (3).

$$\mathcal{E}_q = \{\, e : \tau(e) \in \{\mathcal{I}\to\mathcal{C}, \mathcal{C}\to\mathcal{I}\},\ q(e) < q \,\} \ \cup\ \{\, e : \tau(e) \notin \{\mathcal{I}\to\mathcal{C}, \mathcal{C}\to\mathcal{I}\} \,\} \tag{3}$$

The edge set $\mathcal{E}_q$ contains only the investment edges whose quarters are strictly earlier than q, together with all static edges. Investment relations from quarter q onward are therefore invisible when quarter q is forecast.

Financing rounds do not enter the graph as nodes. Each financing round r is a query object, represented by the node $c(r)$ of its startup and by the attributes of the round. This representation is given by Equation (4).

$$r = \big(c(r),\ \text{type}(r),\ \text{amount}(r),\ q(r)\big) \tag{4}$$

In Equation (4), $\text{type}(r)$ denotes the round type, $\text{amount}(r)$ denotes the amount raised, and $q(r)$ denotes the quarter of the round. All graph structure information of a financing round is obtained through $c(r)$.

The task is defined as follows. Given the snapshot graph G_q and the set $\mathcal{R}_q$ of financing rounds in quarter q, the model produces a ranking over the set of candidate investors for each $r \in \mathcal{R}_q$.

3.2. Capacity-Constrained Allocation

For each quarter q, the ranking task of Section 3.1 is cast as an allocation of the financing rounds in $\mathcal{R}_q$ to the candidate investors in $\mathcal{I}$. The decision variable is the allocation matrix $P \in \mathbb{R}_+^{|\mathcal{R}_q| \times |\mathcal{I}|}$, whose entry P_{ri} is the mass of quota that financing round r assigns to investor i. The financing round demand a_r is the syndicate prior of its round type, i.e., the mean number of co-investors of training rounds of the same type, as given in Equation (5).

$$P \in \mathbb{R}_+^{|\mathcal{R}_q| \times |\mathcal{I}|}, \qquad a_r = \bar{k}_{\text{type}(r)}, \qquad \bar{k}_{\text{seed}} = 1.812, \ \bar{k}_{\text{angel}} = 2.136, \ \bar{k}_{\text{venture}} = 2.433. \tag{5}$$

The demand a_r thus fixes how many investor slots each financing round offers in the allocation.

The investment capacity b_i is the mean number of financing rounds that investor i joined per quarter over the four quarters before q, with a floor of 0.25, as written in Equation (6).

$$b_i = \max\left\{\tfrac{1}{4}\sum_{j=1}^{4} n_i(q-j),\ 0.25\right\}. \tag{6}$$

The count $n_i(\cdot)$ includes only financing rounds that occurred before quarter q, so b_i follows the causal quarterly snapshot of Section 3.1 and uses no information from the current quarter. The quantity b_i is an activity-based proxy for the number of financing rounds an investor can enter in a quarter. Fund size, available capital and staffing are not in the data, so b_i measures recent deal activity only, and capacity in the results is read in this activity sense. Section 5.5 reports results under other window lengths, other floors and an exponentially weighted activity measure.

With a four-quarter window, an investor with exactly one deal in the window has $b_i = 0.25$. The floor therefore gives an investor without any deal in the window the capacity of an investor with one deal per year. The floor also keeps every column marginal strictly positive, which the log-domain Sinkhorn iteration requires because $\log b_i$ must be finite. Inactive investors consequently retain a small nonzero share of the allocation.

In the first test quarter (2014Q3), 4763 candidate investors face 2023 financing rounds. The capacity b_i in this quarter has a mean of 0.998, a median of 0.250 and a maximum of 56.5, and 56.5% of the investors sit at the floor of 0.25. These investors account for 9.2% of the observed investment edges of the quarter. Across the six test quarters, the share of investors at the floor ranges from 56.5% to 64.8%, and their share of the observed investment edges ranges from 8.3% to 10.7%. The floor thus covers the majority of investors but only a small fraction of the observed deals.

The feasible set under the capacity constraint consists of the nonnegative matrices that satisfy two groups of marginal constraints, as given in Equation (7).

$$\mathcal{P}(a,b) = \{P \in \mathbb{R}_+^{|\mathcal{R}_q| \times |\mathcal{I}|} \ : \ P\mathbf{1} \le a, \ \ P^\top \mathbf{1} \le b\}. \tag{7}$$

The row constraint $P\mathbf{1} \le a$ requires that the quota allocated by each financing round does not exceed its demand. The column constraint $P^\top \mathbf{1} \le b$ requires that the quota received by each investor does not exceed its investment capacity. When the total demand $\mathbf{1}^\top a$ differs from the total capacity $\mathbf{1}^\top b$, the two groups of constraints cannot hold with equality at the same time, and the inequality form keeps the feasible set nonempty.

Given the matching score matrix S produced by the encoder of Section 4, the quarterly matching is the linear program in Equation (8).

$$P^\star = \arg\max_{P \in \mathcal{P}(a,b)} \langle S, P \rangle. \tag{8}$$

The optimal plan $P^\star$ places quota on high-scoring pairs as far as demand and capacity permit.

The main model uses the equality-normalized form, in which the investment capacities are rescaled by the total demand to $\tilde{b}$ so that the column constraints hold with equality, as given in Equation (9).

$$\tilde{b} = \frac{\mathbf{1}^\top a}{\mathbf{1}^\top b}\, b, \qquad \Pi(a, \tilde{b}) = \{P \geq 0 : \; P\mathbf{1} = a, \; P^\top \mathbf{1} = \tilde{b}\}. \tag{9}$$

Because $\mathbf{1}^\top \tilde{b} = \mathbf{1}^\top a$, the set $\Pi(a, \tilde{b})$ is the special case of $\mathcal{U}(a, \tilde{b})$ in which both groups of inequalities hold with equality. The linear program (8) over this constraint set, with marginals a and $\tilde{b}$, is the problem to which Section 3.3 adds the entropy term.

3.3. Entropic Regularization and Sinkhorn Iterations

Adding an entropy term to the linear program of Section 3.2 turns the objective into an entropic optimal transport problem:

$$\max_P \langle S, P \rangle + \varepsilon H(P), \qquad H(P) = -\sum_{r,i} P_{ri}\big(\log P_{ri} - 1\big) \tag{10}$$

In Equation (10), S is the matching score matrix and $\varepsilon > 0$ is the regularization strength. The entropy term $H(P)$ is strictly concave on the nonnegative orthant. The objective is therefore strictly concave on the convex feasible set defined in Section 3.2, and the optimal transport plan is unique.

The optimum takes the form of a kernel matrix scaled on both sides:

$$P^\star = \mathrm{diag}(u)\, K\, \mathrm{diag}(v), \qquad K = \exp(S/\varepsilon) \tag{11}$$

In Equation (11), K is the kernel matrix obtained by the element-wise exponential. The scaling vectors u and v correspond to the marginal constraints on the financing round demand side and the investment capacity side, respectively. They are related to the dual potentials f and g through $u = \exp(f/\varepsilon)$ and $v = \exp(g/\varepsilon)$.

The Sinkhorn iteration computes the two scaling vectors alternately:

$$u \leftarrow a \oslash (Kv), \qquad v \leftarrow b \oslash (K^\top u) \tag{12}$$

In Equation (12), $\oslash$ denotes element-wise division, a is the financing round demand, and b is the investment capacity. Each update makes the current transport plan satisfy the marginal constraints on one side exactly. Alternating the two updates drives the plan toward satisfying the constraints on both sides simultaneously.

Our implementation performs the same iteration in the log domain:

$$f \leftarrow \varepsilon \log a - \varepsilon \log\Big(\sum_i \exp\big((S_{ri} + g_i)/\varepsilon\big)\Big) \tag{13}$$

Equation (13) gives the update of the dual potential f, and the update of g is symmetric to it. The log-domain form is needed because $\exp(S/\varepsilon)$ underflows when ε is small, so scaling directly by Equation (12) would lose numerical accuracy. This log-domain iteration and the resulting transport plan P are what the model in Section 4 computes.

4. Methodology

4.1. Model Overview

As shown in Figure 1, CapTHG consists of three parts, namely the relation-specific heterogeneous attention encoder on the causal quarterly snapshot, the capacity-constrained Sinkhorn layer and the joint training objective. The encoder outputs the matching score matrix S. The Sinkhorn layer converts S into a capacity-feasible transport plan P. The training objective acts on the scores and on the plan simultaneously.

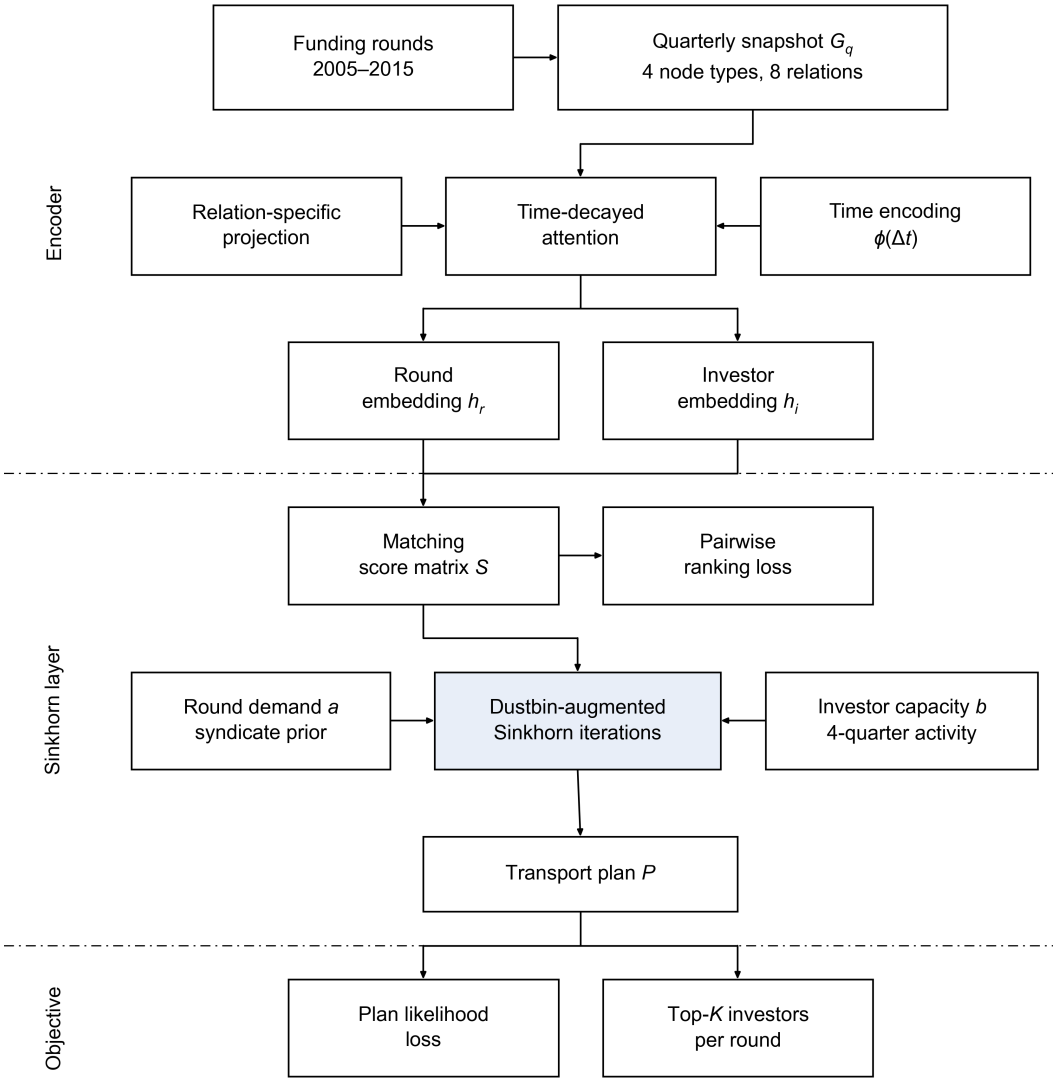


Figure 1. Overview of CapTHG: the encoder on a causal quarterly snapshot produces the matching score matrix, and the Sinkhorn layer turns it into a capacity-feasible transport plan. Solid arrows indicate the data flow; dash-dotted lines separate the three parts of the model (encoder, Sinkhorn layer and training objective, labeled on the left); the shaded box marks the dustbin-augmented Sinkhorn iterations, the core of the capacity-constrained layer, and all other boxes are unshaded.

On the causal quarterly snapshot G_q, the encoder aggregates the four node types through relation-specific heterogeneous attention and produces representations of financing rounds and investors. The inner product of these two representations gives the matching score matrix S (Section 4.2). The Sinkhorn layer feeds S, the financing round demand a and the investment capacity b into log-domain Sinkhorn iterations and outputs a transport plan P that satisfies the capacity constraints. In the main form, the column marginals are equality-normalized according to the investment capacity. The dustbin-augmented form covers the general case in which the investment capacity is an upper bound. Section 4.3 gives the details of the layer, and Section 4.4 establishes its properties. In the training stage, the pairwise ranking loss and the plan likelihood loss are optimized jointly. In the inference stage, the top K investors of each round are taken in the row order of the transport plan (Section 4.5). The description of the model begins with the input features of the encoder in Section 4.2.

4.2. Relation-Specific Attention Encoder

The input features $x^{(0)}$ of the encoder are constructed differently for each node type. Investors, sectors and regions each have a learnable embedding table, and the input feature of such a node is the corresponding row of its table. Startups carry no identity embedding. Their input features are instead composed of a shared base vector and the mean of their attribute embeddings, as given in Equation (14).

$$x_c^{(0)} = b + \frac{1}{|\mathcal{S}(c)|} \sum_{s \in \mathcal{S}(c)} e_s + \frac{1}{|\mathcal{R}(c)|} \sum_{k \in \mathcal{R}(c)} e_k \tag{14}$$

Here, b is the base vector shared by all startups, $\mathcal{S}(c)$ and $\mathcal{R}(c)$ are the sets of sector nodes and region nodes to which startup c belongs, and e_s and e_k are the corresponding sector embedding and region embedding. With this construction, a startup that first appears in a snapshot can be represented without training any parameters of its own, so the encoder is inductive. All representations have dimension 64.

Each layer assigns projection matrices according to relation type. For every edge $(u \to v) \in \mathcal{E}_q$ in the causal quarterly snapshot, the relation type t of the edge selects a relation-specific key projection W_t^K and value projection W_t^V. The query of the target node is obtained through the shared query projection W^Q. The attention logit is then computed by Equation (15).

$$e_{uv} = \frac{\langle W_t^K x_u,\, q_v \rangle}{\sqrt{d_h}} + \beta_t, \qquad q_v = W^Q x_v \tag{15}$$

In this equation, d_h is the dimension of each attention head. The term β_t is the learnable prior scalar of relation type t, which captures the baseline importance of that relation in aggregation.

The attention weights are normalized jointly over all incoming edges of the target node. For a given target node v, the logits of all its incoming edges enter a single softmax regardless of relation type. The normalized weights then form the message vector as a weighted sum, as expressed in Equation (16).

$$\alpha_{uv} = \frac{\exp(e_{uv})}{\sum_{u'} \exp(e_{u'v})}, \qquad m_v = \sum_u \alpha_{uv} W_t^V x_u. \tag{16}$$

Each layer uses 2 attention heads. The message vectors of the heads are concatenated before the update.

The layer output is formed by a residual connection followed by layer normalization. As shown in Equation (17), the aggregated message passes through the output projection W^O and a GELU nonlinearity and is then added to the output of the previous layer.

$$x_v^{(l+1)} = \mathrm{LayerNorm}\big(x_v^{(l)} + \mathrm{GELU}(W^O m_v)\big). \tag{17}$$

The encoder stacks 2 attention layers, so the receptive field of each node covers its two-hop neighborhood. Temporal information enters only through the snapshot itself, because $\mathcal{E}_q$ contains only investment edges dated before quarter q. The model therefore applies no additional time weighting to individual edges.

The representation of a financing round r is produced by a multilayer perceptron (MLP). Its input concatenates the final-layer state of the startup, the round-type embedding, the log-scaled amount $\tilde{a}(r)$ of the round, and a missing-amount indicator. The investor representation is a linear projection of the final-layer state of the investor. The matching score is the scaled inner product of the two representations, as defined in Equation (18).

$$h_r = \mathrm{MLP}\big([\, x_{c(r)}^{(L)};\, \mathrm{emb}(\mathrm{type}(r));\, \tilde{a}(r);\, \mathbf{1}_{\text{amount missing}}\,]\big), \quad h_i = W^I x_i^{(L)}, \quad S_{ri} = \frac{\langle h_r, h_i \rangle}{\sqrt{d}}. \tag{18}$$

When the amount is missing, $\tilde{a}(r)$ is set to 0 and the indicator marks the absence. The symbol d denotes the representation dimension.

4.3. Capacity-Constrained Sinkhorn Layer

The matching score matrix S produced by the encoder has no fixed scale. To make the value of the regularization strength ε independent of the score scale, we first standardize all elements of S:

$$\tilde{S} = \frac{S - \mu_S}{\sigma_S}, \qquad \mu_S = \frac{1}{mn}\sum_{r,i} S_{ri}, \quad \sigma_S = \Big(\frac{1}{mn}\sum_{r,i}(S_{ri} - \mu_S)^2\Big)^{1/2} \tag{19}$$

The quantities μ_S and σ_S in Equation (19) are computed quarter by quarter, so the standardized scores $\tilde{S}$ of snapshots from different quarters lie on a common scale.

The main form rescales the investment capacity to the total financing round demand, so that the column constraint holds with equality. With the rescaled capacity $\tilde{b}$, the two marginals are

$$P\mathbf{1} = a, \qquad P^\top \mathbf{1} = \tilde{b}, \qquad \tilde{b} = b\,\frac{\mathbf{1}^\top a}{\mathbf{1}^\top b} \tag{20}$$

The two sets of marginals in Equation (20) have the same total, so the feasible set is nonempty. In this equality-normalized form, the quota of each investor is proportional to the share of its investment capacity in the total capacity. The form therefore uses only the ratios $b_i/\mathbf{1}^\top b$.

In the general case, the investment capacity is an upper bound and the total capacity may differ from the total demand. The dustbin-augmented form handles this case by appending a dustbin row and a dustbin column to $\tilde{S}$, all of whose elements take the dustbin score α. This gives the augmented matrix

$$\bar{Z} = \begin{pmatrix} \tilde{S} & \alpha\mathbf{1}_m \\ \alpha\mathbf{1}_n^\top & \alpha \end{pmatrix} \tag{21}$$

The dustbin score α is a single learnable scalar of the model. It is initialized at 0 and updated by backpropagation together with the encoder parameters through the plan likelihood loss. The augmented transport plan $\bar{P}$ associated with Equation (21) has the marginals

$$\bar{P}\mathbf{1} = \begin{pmatrix} a \\ \mathbf{1}^\top b \end{pmatrix}, \qquad \bar{P}^\top \mathbf{1} = \begin{pmatrix} b \\ \mathbf{1}^\top a \end{pmatrix} \tag{22}$$

Both sides of Equation (22) sum to $\mathbf{1}^\top a + \mathbf{1}^\top b$. Unmatched demand flows into the dustbin column and unused investment capacity flows into the dustbin row, so the real block satisfies $P\mathbf{1} \le a$ and $P^\top \mathbf{1} \le b$. The main form does not contain α, so α plays a role only in the dustbin variant reported in Section 5.3.

Both forms are solved by the log-domain Sinkhorn iteration. The dual potentials f and g are updated alternately by the rule of Equation (13), in which S is replaced by $\bar{Z}$ (by $\tilde{S}$ in the main form). After T iterations, the plan is recovered as

$$\log P = \big(\bar{Z} + f \oplus g\big)/\varepsilon, \qquad (f \oplus g)_{ri} = f_r + g_i \tag{23}$$

The transport plan is the real block of this matrix. For the main form, $\bar{Z}$ in Equation (23) is replaced by $\tilde{S}$ and the marginals by those of Equation (20). We set $\varepsilon = 0.1$ and run 20 Sinkhorn iterations in training and 50 in inference. The investment capacity b is the mean number of deals of the investor in the last four quarters, with a floor of 0.25, as defined in Section 3.2.

Each Sinkhorn iteration is a logsumexp step, so the layer is a finite composition of logsumexp steps. The transport plan is therefore differentiable with respect to $\tilde{S}$ and, in the dustbin-augmented form, with respect to α. Gradients thus pass to the encoder by ordinary backpropagation, and Proposition 2 in Section 4.4 gives their explicit form.

During training, each quarterly problem is approximated by mini-batches of financing rounds. The training data contain 32 training quarters, of which 21 contain more than 768 training rounds; these 21 quarters hold 79.2% of the training rounds, and the largest contains 2052. A quarter with at most 768 rounds is solved whole. For a larger quarter q with round set $\mathcal{R}_q$, the batch B is a uniform random sample of $|B| = 768$ of its rounds, drawn anew in every epoch, and every capacity is multiplied by $|B|/|\mathcal{R}_q|$. The total capacity of the batch then matches the expected total demand of the batch. The batch problem is thus the quarterly problem with every investor's capacity shrunk by the sampling fraction, and its column marginals keep the proportions of the quarter. The rows of a batch see the same investors and the same relative capacities as in the full quarter, so the batch plan approximates the quarterly plan restricted to the sampled rows. Only the demand mix of the sampled rounds changes from batch to batch. Inference never uses batches, and the test plan is solved on all rounds of the quarter. Section 5.5 compares batch sizes of 384, 768 and 1536 with whole quarters and measures how close the batch plans are to the full-quarter plan. The properties of the augmented problem and of the layer are established in Section 4.4.

4.4. Theoretical Properties

Proposition 1. *Let $\alpha \in \mathbb{R}$, and let the augmented matrix $\bar{Z} \in \mathbb{R}^{(m+1)\times(n+1)}$ contain the matching score matrix S in its real block and the dustbin score α in all remaining entries. Let the marginal constraints consist of the row marginal $(a, \mathbf{1}^\top b)$ and the column marginal $(b, \mathbf{1}^\top a)$. Then the real block P of any feasible solution $\bar{P} \geq 0$ of the augmented problem satisfies $P\mathbf{1} \leq a$ and $P^\top \mathbf{1} \leq b$. Conversely, every $P \geq 0$ satisfying these two inequalities is the real block of a unique $\bar{P}$, whose slack column is $s = a - P\mathbf{1}$, whose slack row is $t = b - P^\top \mathbf{1}$ and whose corner element is $\mathbf{1}^\top P\mathbf{1}$, and the augmented objective satisfies Equation (24).*

Proof. Nonnegativity and the row marginal give $P\mathbf{1} + s = a$ with $s \geq 0$; hence $P\mathbf{1} \leq a$. In the same way, the column marginal gives $P^\top \mathbf{1} \leq b$. Conversely, let such a P be given. The row marginal then determines the slack column uniquely as $s = a - P\mathbf{1}$, and the column marginal determines the slack row uniquely as $t = b - P^\top \mathbf{1}$. The marginal of the last row fixes the corner element at $m_{\text{corner}} = \mathbf{1}^\top b - \mathbf{1}^\top t = \mathbf{1}^\top P\mathbf{1}$, and the marginal of the last column fixes it at $\mathbf{1}^\top a - \mathbf{1}^\top s = \mathbf{1}^\top P\mathbf{1}$. Since the two marginals yield the same value, they are compatible, and $\bar{P}$ is unique. Splitting $\langle \bar{Z}, \bar{P} \rangle$ into the real block, the slack row, the slack column and the corner element gives $\langle S, P \rangle + \alpha(\mathbf{1}^\top s + \mathbf{1}^\top t + m_{\text{corner}})$. Substituting s, t and m_{corner} and rearranging yields the identity

$$\langle \bar{Z}, \bar{P} \rangle = \langle S - \alpha \mathbf{1}\mathbf{1}^\top,\ P \rangle + \alpha(\mathbf{1}^\top a + \mathbf{1}^\top b) \tag{24}$$

The second term on the right-hand side of Equation (24) does not depend on P, so the dustbin score α acts as a retention threshold on the real block: only pairs whose score exceeds α increase the objective value. □

Theorem 1. *Let $\varepsilon > 0$, and let the entropy be $H(\bar{P}) = -\sum_{ij} \bar{P}_{ij}(\log \bar{P}_{ij} - 1)$. Let P_ε be the unique optimal solution of the entropic regularization problem over the marginal constraint set, and let* OPT *be the optimal value of the linear program $\max_{\bar{P}} \langle \bar{Z}, \bar{P} \rangle$ over the same constraint set. Denote the total mass by $M = \mathbf{1}^\top a + \mathbf{1}^\top b$ and the size of the augmented matrix by $N = (m+1)(n+1)$. Then, Equation (25) holds; hence $\langle \bar{Z}, P_\varepsilon \rangle \to$ OPT as $\varepsilon \to 0$.*

Proof. The left inequality follows directly, because P_ε is feasible and OPT is the maximum over the feasible set. Let $P^\star$ denote an optimal solution of the linear program. Since P_ε is optimal for the entropic regularization objective, comparing the two solutions under this objective gives $\langle \bar{Z}, P_\varepsilon \rangle + \varepsilon H(P_\varepsilon) \geq \mathrm{OPT} + \varepsilon H(P^\star)$. Rearranging gives $\mathrm{OPT} - \langle \bar{Z}, P_\varepsilon \rangle \leq \varepsilon \left(H(P_\varepsilon) - H(P^\star) \right)$. The entropy of a nonnegative matrix with total mass M and N entries obeys the standard bounds $-M \log M \leq H \leq M \log(N/M)$. Hence, the difference of the two entropies does not exceed $M \log N$, which gives the right inequality:

$$0 \leq \mathrm{OPT} - \langle \bar{Z}, P_\varepsilon \rangle \leq \varepsilon M \log N \tag{25}$$

Equation (25) shows that the regularization bias depends linearly on ε. □

Theorem 2. *Let $\varepsilon > 0$, and let the kernel matrix $K = \exp(\bar{Z}/\varepsilon)$ be entrywise positive. Then, each half-step update of the Sinkhorn iteration is a contraction mapping in the Hilbert projective metric $d_{\mathcal{H}}$, and its contraction ratio κ and the projective diameter Δ of K are given by Equation (26). Hence, the dual potentials converge linearly at rate κ^2, and the marginal error after T iterations decays as $O(\kappa^{2T})$.*

Proof. The theorem is a standard application of the Birkhoff–Hopf contraction theorem to positive matrix scaling; we do not reprove that theorem and only show how it applies. Since K is entrywise positive, it maps the positive cone into itself, and its projective diameter Δ is finite. By the standard result, one application of K contracts $d_{\mathcal{H}}$ with ratio $\tanh(\Delta/4)$. Row normalization and column normalization act as positive scalings, which leave the distance in the projective metric unchanged. One round of updates consists of one action in the row direction and one in the column direction. Since K and $K^\top$ have the same projective diameter, the dual potentials converge linearly at rate κ^2, and the decay rate of the marginal error follows:

$$\kappa = \tanh(\Delta/4) < 1, \qquad \Delta = \sup_{i,j,k,l} \log \frac{K_{ij} K_{kl}}{K_{il} K_{kj}} \tag{26}$$

The diameter Δ in Equation (26) increases as ε decreases, so a smaller ε yields slower convergence, consistent with the numerical results of Section 5.6. □

Proposition 2. *Let $\varepsilon > 0$, let Π be the marginal constraint set given in Section 4.3, let $W_\varepsilon(\bar{Z}) = \max_{\bar{P} \in \Pi} \langle \bar{Z}, \bar{P} \rangle + \varepsilon H(\bar{P})$, and let $\bar{P}^\star$ be the transport plan attaining the maximum. Then W_ε is differentiable in $\bar{Z}$, and the envelope identity in Equation (27) holds.*

Proof. The set Π is a compact convex set, and the objective is strictly concave in $\bar{P}$; hence the optimal solution $\bar{P}^\star$ exists and is unique. The objective is linear in $\bar{Z}$, and its partial derivative with respect to $\bar{Z}$ equals $\bar{P}$. By Danskin's theorem, the value function is differentiable wherever the optimal solution is unique, and its gradient equals the partial derivative evaluated at the optimal solution, which is Equation (27):

$$\frac{\partial W_\varepsilon(\bar{Z})}{\partial \bar{Z}} = \bar{P}^\star \tag{27}$$

Equation (27) shows that the gradient of the layer is the transport plan itself, which is the basis for end-to-end backpropagation of the plan likelihood loss of Section 4.5. Numerical checks of the four results above are reported in Section 5.6. □

4.5. Training Objective and Inference

The first training objective is the pairwise ranking loss, which acts directly on the order of the entries within each row of the matching score matrix S. For each observed pair

$(r, i) \in O_q$ of training quarter q, that is, a financing round r and an investor i that actually took part in it, 8 negative investors j are sampled uniformly from the candidate investors of that quarter and form the set $\mathcal{N}_{ri}$. Adopting the pairwise ranking loss of Rendle et al. [36], we obtain Equation (28):

$$\mathcal{L}_{\mathrm{BPR}} = -\frac{1}{8|O_q|} \sum_{(r,i)\in O_q} \sum_{j\in\mathcal{N}_{ri}} \log \sigma\big(S_{ri} - S_{rj}\big) \tag{28}$$

where $\sigma(\cdot)$ is the sigmoid function.

The second training objective is the plan likelihood loss, which places a supervision signal on the transport plan P solved for the same quarter. For each observed pair, the loss takes the negative logarithm of the pair's share of the round's mass in financing round r, as given in Equation (29):

$$\mathcal{L}_{\mathrm{plan}} = -\frac{1}{|O_q|} \sum_{(r,i)\in O_q} (\log P_{ri} - \log a_r) \tag{29}$$

Here, a_r is the financing round demand of financing round r. Minimizing this term assigns as much of each round's mass as possible to its observed investors. By Proposition 2, the gradient with respect to the matching scores is back-propagated through the transport plan, so the capacity-constrained Sinkhorn layer is trained end to end.

The joint objective is the sum of the two losses, as in Equation (30):

$$\mathcal{L} = \mathcal{L}_{\mathrm{BPR}} + \lambda\, \mathcal{L}_{\mathrm{plan}} \tag{30}$$

with weight $\lambda = 0.5$.

Training proceeds quarter by quarter, with one forward pass and one parameter update per causal quarterly snapshot. The representation dimension is 64, and the encoder has 2 layers with 2 attention heads. During training, the Sinkhorn layer runs 20 iterations with entropic regularization coefficient $\varepsilon = 0.1$. The parameters are optimized with Adam at a learning rate of 3×10^{-3}, with a weight decay of 1×10^{-6} and a gradient clipping threshold of 5.0. Each quarterly batch contains 768 financing rounds. Training runs for at most 40 epochs. Recall@20 is evaluated on the validation quarters every 2 epochs, and early stopping is triggered after 4 consecutive evaluations without improvement.

Inference in a test quarter is carried out in three steps. First, the encoder computes the matching score matrix S on the causal quarterly snapshot of that quarter. Second, the investment capacity of each investor is estimated from its activity in the previous four quarters, and the financing round demand of each financing round is set by its round type. Third, the transport plan P is solved with 50 Sinkhorn iterations. The recommendation list of a round consists of the K investors with the largest entries in that round's row of the plan. The row and column constraints hold for the continuous transport plan P up to the Sinkhorn tolerance. The top-K lists are a rounding of this plan and are not themselves required to respect each investor's capacity exactly. The overload that remains in the lists is measured by the capacity overload ratio OR@10, and Section 5.4 reports it together with a capacity-feasible variant of the lists. The computational cost of these training and inference steps is analysed in Section 4.6.

4.6. Complexity

The snapshot of quarter q contains the node set $\mathcal{V}$ and the edge set $\mathcal{E}_q$. The set of financing rounds is $\mathcal{R}_q$, the candidate investor set is $\mathcal{I}$, and the representation dimension is d.

In each encoder layer, every edge undergoes one constant-dimensional projection and one attention computation, which costs $O(|\mathcal{E}_q|d)$. With two layers, this cost is doubled. The node update costs $O(|\mathcal{V}|d^2)$. The matching score matrix is formed from inner products

between the representations of the two sides, at a cost of $O(|\mathcal{R}_q||\mathcal{I}|d)$. Each iteration of the capacity-constrained Sinkhorn layer consists of two logsumexp passes and costs $O(|\mathcal{R}_q||\mathcal{I}|)$. Running T iterations therefore costs $O(T|\mathcal{R}_q||\mathcal{I}|)$. The memory cost is of the same order and is dominated by storing the score matrix and the transport plan.

With 4763 candidate investors and about 2023 financing rounds in a single test quarter, the matching score matrix and the transport plan are dense matrices with about ten million entries each. The measured running times are reported in Section 5.7.

5. Experiments

5.1. Experimental Setup

Datasets: The experimental data are taken from the public Crunchbase snapshot. We retain the seed, angel and venture rounds from January 2005 to December 2015. Candidate investors are the investors with at least three deals before 2014, 4763 in total. The rule follows from the model. An investor's representation is a learned embedding, and its capacity is an average of past deals, so every investor needs a deal history before the test period. Three deals before 2014 give every candidate such a history in the training period. Over the test period from 2014Q3 to 2015Q4, with the same round types, candidate investors take part in 59.1% of all investment edges in the raw snapshot. Among the test-period rounds, 72.8% contain at least one candidate investor, and these 9679 rounds form the evaluated test set. Investors with no deal before 2014 number 7166 of the 11,767 investors active in the test period. They account for 28.8% of the test-period investment edges and lie outside the candidate set. Section 5.5 repeats the evaluation with the threshold lowered to two deals and to one deal.

The temporal heterogeneous graph built from these records contains 30,963 startups, 834 sector nodes, 905 region nodes, 48,921 financing rounds and 108,272 investment edges. Table 1 lists these counts together with the edge counts of each relation type. The test period contains 20,902 investment edges. The mean syndicate size of the test rounds is 2.16, so syndication is the norm rather than the exception.

The data are divided by time rather than at random. The training set consists of the 35,162 financing rounds before 2014. The validation set consists of the 4080 financing rounds in 2014Q1–Q2. The test set consists of the 9679 financing rounds in the six quarters from 2014Q3 to 2015Q4. Evaluation proceeds by quarterly rolling evaluation. When the model predicts the financing rounds of quarter q, it encodes only the causal quarterly snapshot, which contains the investment edges dated before quarter q. Information on future deals therefore enters neither the node representations nor the matching scores.

Table 1. Statistics of the temporal heterogeneous graph built from the Crunchbase snapshot (seed, angel and venture rounds, 2005–2015; candidate investors have at least three deals before 2014).

Item	Count
Candidate investors	4763
Startups	30,963
Sector nodes	834
Region nodes	905
Funding rounds (train/validation/test)	35,162/4080/9679
Investment edges (train/validation/test)	78,733/8637/20,902
Startup–sector edges	73,987
Startup–region edges	28,040
Investor–region edges	3824
Mean syndicate size of test rounds	2.16

Evaluation Metrics: Matching quality is measured by five ranking metrics, namely Recall@10, Recall@20, NDCG@10, NDCG@20 and HR@10. Capacity behavior is measured by three quarter-level metrics. The capacity overload ratio OR@10 is the share of top-10 exposure that exceeds each investor's capacity share in a quarter, and a lower value means that the recommendations respect the capacity constraints more closely. The investor coverage Cov@10 is the share of candidate investors that appear in at least one top-10 list. The exposure Gini coefficient Gini@10 is the Gini coefficient of top-10 exposure and indicates whether exposure concentrates on a few investors.

Baselines: We compare our model with seven baseline models. Popularity ranks investors by their number of deals in the past four quarters. SR-Profile scores investors by their historical sector and region profile. FM-BPR is a factorization model without graph propagation. LightGCN is a graph collaborative filtering model with simplified propagation. R-GCN aggregates neighbors by relation type. HGT is a Transformer encoder for heterogeneous graphs. TGAT adds time encoding to attention to capture graph evolution.

Qi et al. [66] show that the ranking of explanations produced by a trained model need not survive retraining, and they make its reproducibility across bootstrap refits an explicit optimization objective; for this reason, every learned result in this study is obtained from several independent training runs. The number of seeds depends on the experiment group. The main comparison in Table 2 and the capacity and exposure comparison use five seeds for every learned model. The ablation study uses five seeds for every variant, and the comparison of capacity-window lengths also uses five seeds. Hyperparameter sensitivity (λ, ε, dimension and number of layers), capacity floors, exponentially weighted capacity, mini-batch sizes and candidate thresholds use three seeds. Popularity and SR-Profile are deterministic. We report the mean and standard deviation over seeds.

Table 2. Test results of the quarterly rolling evaluation (2014Q3–2015Q4, 9679 funding rounds, 4763 candidate investors). Learned models: mean ± standard deviation over five seeds; Popularity and SR-Profile are deterministic. Baselines rank by their scores, CapTHG ranks by the Sinkhorn transport plan. Best value in each column in bold.

Model	Recall@10	Recall@20	NDCG@10	NDCG@20	HR@10
Popularity	0.1006	0.1339	0.0540	0.0648	0.1563
SR-Profile	0.1695	0.2353	0.1133	0.1333	0.2744
FM-BPR	0.1138 ± 0.0132	0.1655 ± 0.0200	0.0682 ± 0.0103	0.0838 ± 0.0124	0.1911 ± 0.0193
LightGCN	0.1319 ± 0.0153	0.1919 ± 0.0201	0.0775 ± 0.0095	0.0956 ± 0.0110	0.2144 ± 0.0233
R-GCN	0.2413 ± 0.0079	0.3216 ± 0.0071	0.1626 ± 0.0059	0.1870 ± 0.0057	0.3747 ± 0.0105
HGT	0.2235 ± 0.0072	0.3030 ± 0.0083	0.1523 ± 0.0056	0.1763 ± 0.0061	0.3510 ± 0.0102
TGAT	0.2061 ± 0.0046	0.2882 ± 0.0061	0.1354 ± 0.0048	0.1602 ± 0.0052	0.3269 ± 0.0051
CapTHG	**0.3202 ± 0.0058**	**0.3919 ± 0.0060**	**0.2520 ± 0.0053**	**0.2740 ± 0.0054**	**0.4722 ± 0.0061**

Statistical Tests: The Sinkhorn plan couples all rounds of a quarter through the column constraints, so rounds of the same quarter are not independent observations. The hypothesis tests therefore take the quarter-by-seed cell as the primary unit. A metric is averaged over the test rounds of one quarter for one seed, which yields $6 \times 5 = 30$ cells per model. These cells are compared by the one-sided Wilcoxon signed-rank test, following Demšar's recommendations for comparing learning algorithms [67]. The quarter-level capacity metrics OR@10, Cov@10 and Gini@10 are defined per quarter and use the same cells. Two further checks are reported. The first is an exact one-sided sign test over the 6 seed-averaged quarters, whose smallest attainable p value is $1/64 \approx 0.016$. The second is a 95% cluster-bootstrap interval with 10,000 replicates, in which quarters are resampled with replacement and rounds are resampled within each drawn quarter. The round-level Wilcoxon test over the 9679 test rounds is reported only as a descriptive complement.

Hyperparameter Settings: The hyperparameters follow Section 4.5, with dimension 64, 2 layers, 2 attention heads, $\varepsilon = 0.1$, $\lambda = 0.5$ and 50 Sinkhorn iterations at inference. Under this protocol, Section 5.2 reports the main results of our model and the seven baseline models.

5.2. Main Results

Under the data split, metrics, baselines and seeds defined in Section 5.1, Table 2 reports the recall and the normalized discounted cumulative gain of the eight models in the quarterly rolling evaluation. The test period runs from 2014Q3 to 2015Q4 and covers six test quarters and 9679 financing rounds. For the learned models, each value is the mean over five seeds.

The non-learning and shallow baselines perform worst. Popularity reaches a Recall@20 of 0.1339 and an NDCG@10 of 0.0540, so ranking investors by their past activity alone hardly describes which investor funds which startup. FM-BPR raises Recall@20 to 0.1655, which is still low. SR-Profile attains a Recall@20 of 0.2353 and an HR@10 of 0.2744, the best values among the three shallow baselines. The static profiles of startups and investors therefore carry usable matching information, but profile similarity alone does not recover the investment choices actually made in a quarter.

The graph baselines are clearly stronger, and modeling heterogeneous relations matters most. R-GCN is the strongest baseline, with a Recall@20 of 0.3216 and an NDCG@10 of 0.1626. HGT follows with a Recall@20 of 0.3030 and TGAT with 0.2882. LightGCN, which reduces the graph to a homogeneous bipartite graph, reaches only 0.1919. This ordering shows that encoding the multiple relations among startups, investors, industries and regions brings a larger gain than adding time information alone.

CapTHG is the best model on all five metrics. Its Recall@10 is 0.3202, its Recall@20 is 0.3919, its NDCG@10 is 0.2520, its NDCG@20 is 0.2740, and its HR@10 is 0.4722. The standard deviation of Recall@20 across seeds is 0.0060, so the advantage is not an artifact of random initialization. Compared with the 0.3216 of R-GCN, the Recall@20 of CapTHG is higher by 0.0703 in absolute terms and by 21.9% in relative terms. The NDCG@10 gain over R-GCN is 0.0894 (0.2520 versus 0.1626), which is larger than the Recall@20 gain. The capacity-constrained transport plan thus not only retrieves more of the observed investors but also places them higher in the recommendation list.

Hypothesis H1 is tested with the quarter-by-seed cell as the unit, which gives 6 quarters $\times$ 5 seeds = 30 cells, and CapTHG seed s is paired with R-GCN seed s. CapTHG has the higher Recall@20 in all 30 cells, and the one-sided Wilcoxon signed-rank test gives $p = 9.3 \times 10^{-10}$. After averaging over seeds, CapTHG is higher in all 6 test quarters, with an exact one-sided sign test $p = 0.016$. The 95% cluster-bootstrap interval of the Recall@20 gain, obtained by resampling quarters and then rounds within quarters, is $[0.064, 0.077]$ and excludes zero. Every CapTHG seed lies above every R-GCN seed, and the one-sided Mann–Whitney test over the five training runs gives $p = 0.004$. The round-level Wilcoxon test over the 9679 financing rounds also yields $p < 0.001$. H1 therefore holds.

Figure 2 shows that the advantage holds in every test quarter. The quarterly Recall@20 of CapTHG rises from 0.4057 in 2014Q3 to 0.4193 in 2014Q4 and then declines quarter by quarter to 0.3315 in 2015Q4. Over the same period, the value of R-GCN moves from 0.3422 to 0.2620, so CapTHG stays above R-GCN throughout.

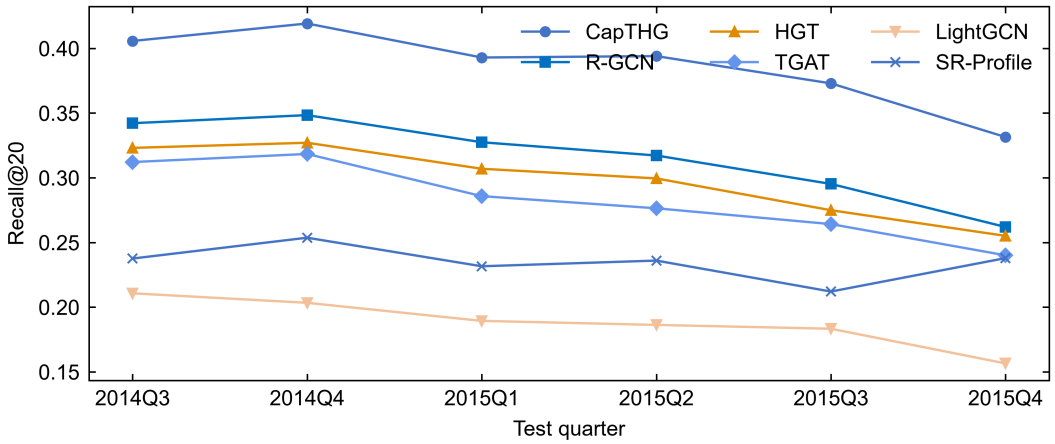


Figure 2. Recall@20 by test quarter.

Two robustness subsets and one alternative accuracy measure support the above conclusion, and Table 3 reports the subsets with the seed variability of every learned model.

Table 3. Robustness subsets of the test period: financing rounds of startups raising their first round (5392 rounds) and the four test quarters of 2015 (5857 rounds). Learned models: mean ± standard deviation over five seeds; Popularity and SR-Profile are deterministic. Baselines rank by their scores, CapTHG ranks by the Sinkhorn transport plan. Best value in each column in bold.

Model	First-Round Startups		2015 Quarters	
	Recall@20	NDCG@10	Recall@20	NDCG@10
Popularity	0.1497	0.0621	0.1321	0.0535
SR-Profile	0.2228	0.1066	0.2289	0.1031
FM-BPR	0.1768 ± 0.0203	0.0716 ± 0.0113	0.1548 ± 0.0184	0.0625 ± 0.0101
LightGCN	0.1955 ± 0.0149	0.0763 ± 0.0080	0.1819 ± 0.0160	0.0700 ± 0.0104
R-GCN	0.2666 ± 0.0066	0.1110 ± 0.0068	0.3063 ± 0.0074	0.1554 ± 0.0064
HGT	0.2424 ± 0.0051	0.0987 ± 0.0037	0.2887 ± 0.0106	0.1460 ± 0.0067
TGAT	0.2541 ± 0.0089	0.1057 ± 0.0076	0.2706 ± 0.0059	0.1270 ± 0.0051
CapTHG	**0.3189 ± 0.0041**	**0.1599 ± 0.0028**	**0.3788 ± 0.0055**	**0.2432 ± 0.0065**

The first subset keeps only the 5392 financing rounds of first-round startups, which form the cold start case. On this subset, CapTHG reaches a Recall@20 of 0.3189 ± 0.0041 against 0.2666 ± 0.0066 for R-GCN, a gain of 0.0523. CapTHG is the best of all models here, ahead of the next best learned models R-GCN (0.2666) and TGAT (0.2541). It is higher in all 30 quarter-by-seed cells ($p = 9.3 \times 10^{-10}$) and in 6 of 6 test quarters ($p = 0.016$), and the cluster-bootstrap interval of the gain is $[0.045, 0.060]$. Its NDCG@10 is 0.1599 ± 0.0028 against 0.1110 ± 0.0068, higher in all 30 cells ($p = 9.3 \times 10^{-10}$).

The second subset uses only the four test quarters of 2015 (5857 financing rounds), whose macro financing environment differs from that of 2014. CapTHG reaches a Recall@20 of 0.3788 ± 0.0055 against 0.3063 ± 0.0074 for R-GCN, a gain of 0.0725 with a cluster-bootstrap interval of $[0.065, 0.080]$. It is higher in all 20 cells ($p = 9.5 \times 10^{-7}$) and in 4 of 4 quarters, where 0.0625 is the smallest attainable exact sign-test p value. Its NDCG@10 is 0.2432 ± 0.0065 against 0.1554 ± 0.0064, higher in all 20 cells ($p = 9.5 \times 10^{-7}$). In both subsets, every CapTHG seed lies above every R-GCN seed (Mann–Whitney $p = 0.004$). When NDCG@10 replaces Recall@20 as the accuracy measure over the full test period, CapTHG reaches 0.2520 against 0.1626 and is higher in all 30 cells ($p = 9.3 \times 10^{-10}$). The conclusion of H1 is therefore unchanged across subsets and measures, and Section 5.3 examines which components of CapTHG produce this gain.

5.3. Ablation Study

To measure the contribution of each component, we conducted an ablation experiment on the 9679 financing rounds of the test period. Only one component was changed at a time, and all other settings were kept identical to the complete model. Every variant was trained with five seeds, and Table 4 reports the mean and standard deviation of each metric.

Table 4. Ablation of CapTHG on the test quarters (mean ± standard deviation over five seeds). OR@10: capacity overload ratio of the top-10 lists (lower is better); Cov@10: investor coverage.

Variant	Recall@20	NDCG@10	OR@10	Cov@10
CapTHG (full)	0.3919 ± 0.0060	0.2520 ± 0.0053	0.3246 ± 0.0203	0.4204 ± 0.0363
w/o Sinkhorn inference	0.3452 ± 0.0095	0.2104 ± 0.0096	0.5486 ± 0.0170	0.3121 ± 0.0363
w/o OT loss	0.2691 ± 0.0071	0.1462 ± 0.0052	0.1686 ± 0.0020	0.8338 ± 0.0251
w/o capacity	0.2171 ± 0.0061	0.1230 ± 0.0055	0.5786 ± 0.0062	0.9170 ± 0.0161
all-history capacity	0.3564 ± 0.0066	0.2313 ± 0.0066	0.3607 ± 0.0209	0.5489 ± 0.0422
w/o heterogeneity	0.3666 ± 0.0077	0.2194 ± 0.0027	0.3138 ± 0.0290	0.4516 ± 0.0730
+time-decayed attention	0.3677 ± 0.0084	0.2240 ± 0.0092	0.3006 ± 0.0562	0.5108 ± 0.0994
+dustbin slack	0.3688 ± 0.0041	0.2308 ± 0.0039	0.3671 ± 0.0310	0.4012 ± 0.0516

When the matching scores are ranked directly at inference, Recall@20 falls from 0.3919 to 0.3452 and NDCG@10 falls from 0.2520 to 0.2104. The capacity overload ratio OR@10 rises from 0.3246 to 0.5486, and investor coverage Cov@10 drops from 0.4204 to 0.3121. The transport plan of the differentiable Sinkhorn layer is therefore the step that carries the capacity constraint into the ranking.

Removing the plan likelihood loss lowers ranking quality sharply. With the pairwise ranking loss alone in training and capacity-constrained Sinkhorn ranking at inference, Recall@20 drops to 0.2691 and NDCG@10 drops to 0.1462. At the same time, OR@10 falls to 0.1686 and Cov@10 rises to 0.8338. The plan likelihood loss is thus the main supervision signal that aligns the encoder with the transport plan.

The lower OR@10 of this variant has a clear mechanism. Trained by the pairwise ranking loss alone, the encoder produces scores that concentrate each round's plan on a few investors. The 10 largest entries of a row carry 0.66 of the round's mass on average, against 0.30 for the complete model, and these investors are largely the same head investors across rounds. The column constraints then move most of this mass to investors with spare capacity, regardless of fit. The resulting lists reach 83% of the candidate investors and push overload below the level of the observed deals. The investment edges actually observed in the test quarters have an overload ratio of 0.424 against the same capacity shares and involve 27.8% of the candidates per quarter. The low overload of this variant is therefore obtained by giving up relevance. With the plan likelihood loss, the scores are trained through the column rescaling, so the plan keeps the observed investors near the top while respecting the capacity shares. The OR@10 of the complete model (0.3246) lies between score ranking (0.5486) and the plan trained with the pairwise ranking loss alone (0.1686), and it is closer to the observed level. Beyond the level of the observed deals, a lower OR@10 does not indicate a better ranking. Section 5.5 shows that a small weight $\lambda = 0.1$ already moves the model most of the way out of the regime of the pairwise ranking loss alone.

Uniform capacities degrade the model most. With uniform column marginals, Recall@20 is only 0.2171, NDCG@10 is only 0.1230, and OR@10 rises to 0.5786. All three values are the worst among the variants, which shows that the capacity term is indispensable for both accuracy and overload.

The estimation of capacity also affects the result. With capacity estimated from the four-quarter activity window, Recall@20 is 0.3919, whereas the all-history average lowers it

to 0.3564 and raises OR@10 from 0.3246 to 0.3607. Taking the quarter-by-seed cell as the unit, the four-quarter activity window gives a higher Recall@20 in all 30 cells (one-sided Wilcoxon signed-rank test, $p = 9.3 \times 10^{-10}$). It is also higher in all 6 seed-averaged quarters (exact sign test, $p = 0.016$), and the 95% cluster-bootstrap interval of the gain is [0.030, 0.041]. H3 therefore holds, and recent activity gives a better capacity estimate than the all-history average.

Removing heterogeneity causes a mild loss. When one shared projection is used for all relation types, Recall@20 is 0.3666, NDCG@10 is 0.2194, and OR@10 is 0.3138. Relation-specific projections thus improve accuracy, but their contribution is smaller than that of capacity and the plan likelihood loss.

Neither of the two additional designs exceeds the complete model. Adding time-decayed attention, which weights single edges by time on top of the causal quarterly snapshot, gives a Recall@20 of 0.3677, an NDCG@10 of 0.2240, and an OR@10 of 0.3006. Adopting the dustbin form, which turns the column marginals into inequalities, gives a Recall@20 of 0.3688, an NDCG@10 of 0.2308, and an OR@10 of 0.3671. Both designs remain below the complete model in Recall@20 and NDCG@10.

Overall, capacity and the plan likelihood loss are the two most critical components, followed by Sinkhorn ranking and heterogeneity. Section 5.4 compares all encoders under score ranking and capacity-constrained Sinkhorn ranking.

5.4. Capacity and Exposure

In Section 5.3, the encoder of CapTHG trained without the plan likelihood loss lost accuracy even though it was still ranked by the Sinkhorn plan. This subsection therefore separates the ranking rule from the encoder for all learned encoders. The capacity overload ratio OR@10 measures how much demand the top-10 lists place on investors whose capacity is already used up. Investor coverage Cov@10 is the share of candidate investors that appear in at least one top-10 list, and the exposure Gini coefficient Gini@10 describes how unevenly top-10 exposure is spread across investors. Table 5 reports these metrics together with Recall@20 for six learned encoders under score ranking (Dot) and under the capacity-constrained Sinkhorn ranking. The Sinkhorn column of every encoder uses the same rule as CapTHG, namely the equality-normalized form with the same regularization strength and number of iterations.

Capacity and exposure improve for all six encoders under the capacity-constrained Sinkhorn ranking. OR@10 falls from 0.7601 to 0.4231 for FM-BPR, from 0.7630 to 0.4509 for LightGCN, from 0.4813 to 0.1730 for R-GCN, from 0.4808 to 0.1689 for HGT, from 0.5084 to 0.1689 for TGAT and from 0.5486 to 0.3246 for CapTHG. Cov@10 rises for every encoder, from 0.3121 to 0.4204 for CapTHG and to between 0.8014 and 0.8377 for R-GCN, HGT and TGAT. Gini@10 falls from 0.9086 to 0.8772 for CapTHG and from 0.8606, 0.8705 and 0.8957 to 0.6848, 0.6628 and 0.6691 for the three graph baselines, so exposure moves from a few head investors to many more investors. H2 is tested on quarter-by-seed cells (6 test quarters $\times$ 5 seeds = 30 cells). The OR@10 of CapTHG is lower under the Sinkhorn ranking in all 30 cells (one-sided Wilcoxon signed-rank test, $p = 9.3 \times 10^{-10}$), and after averaging over seeds it is lower in all 6 quarters (exact one-sided sign test, $p = 0.016$), so H2 holds. Cov@10 of CapTHG is higher in all 30 cells and in all 6 quarters with the same p values.

Recall@20 responds differently. It falls from 0.1655 to 0.1450 for FM-BPR, from 0.1919 to 0.1523 for LightGCN, from 0.3216 to 0.2850 for R-GCN, from 0.3030 to 0.2713 for HGT and from 0.2882 to 0.2414 for TGAT. Only CapTHG gains, from 0.3452 to 0.3919. The Sinkhorn ranking thus improves capacity and exposure for every encoder, whereas it improves Recall@20 only for the encoder trained jointly with the plan.

Table 5. Accuracy and capacity metrics of each learned encoder under score ranking (Dot) and under the capacity-constrained Sinkhorn ranking used by CapTHG (Sinkhorn, equality-normalized form, $\varepsilon = 0.1$, 50 iterations), mean over five seeds. OR@10: overload ratio; Cov@10: share of candidate investors that appear in at least one top-10 list; Gini@10: Gini coefficient of top-10 exposure.

Encoder	Recall@20		OR@10		Cov@10		Gini@10	
	Dot	Sinkhorn	Dot	Sinkhorn	Dot	Sinkhorn	Dot	Sinkhorn
FM-BPR	0.1655	0.1450	0.7601	0.4231	0.0880	0.2595	0.9769	0.9138
LightGCN	0.1919	0.1523	0.7630	0.4509	0.0565	0.2249	0.9844	0.9201
R-GCN	0.3216	0.2850	0.4813	0.1730	0.4115	0.8014	0.8606	0.6848
HGT	0.3030	0.2713	0.4808	0.1689	0.3908	0.8377	0.8705	0.6628
TGAT	0.2882	0.2414	0.5084	0.1689	0.3321	0.8200	0.8957	0.6691
CapTHG	0.3452	0.3919	0.5486	0.3246	0.3121	0.4204	0.9086	0.8772

The five baselines are trained with the pairwise ranking loss alone, so their scores never see the column constraint. Under the row-normalized balanced plan, the 10 largest entries of a round carry 0.63 of its mass for R-GCN against 0.30 for CapTHG. The R-GCN scores thus concentrate each round on a few investors, many of them the same head investors across rounds, and the column constraints must move a large part of that mass to other investors. The resulting plans spread exposure far beyond the pattern of the real deals in the test quarters. Applying the same overload formula to the observed investment edges with the same capacity shares gives an overload ratio of 0.424 on average, and the observed syndicates involve 27.8% of the candidate investors per quarter. The graph baselines instead reach Cov@10 of 0.80–0.84 and OR@10 of about 0.17, so many list positions go to investors chosen for spare capacity rather than fit, and Recall@20 falls. CapTHG is trained with the plan likelihood loss (Section 4.5) through the same layer, so its scores already account for the column rescaling. Its plan stays closer to the observed pattern (OR@10 0.3246, Cov@10 0.4204), and Recall@20 rises. The variant w/o OT loss in Section 5.3 shows the same pattern within one encoder, with Recall@20 of 0.2691 and OR@10 of 0.1686.

The three panels of Figure 3 show a comparison encoder by encoder. For all six encoders, OR@10 falls and Cov@10 rises when score ranking is replaced by the capacity-constrained Sinkhorn ranking, so the gains in capacity and exposure come from the ranking rule itself.

The mechanism lies in the column marginals of the transport plan. The capacity-constrained Sinkhorn ranking holds the column marginal of each investor to its capacity share, and the demand in excess of that share is reassigned to investors with spare capacity. Overload therefore decreases and coverage expands, while financing rounds with high matching scores keep priority.

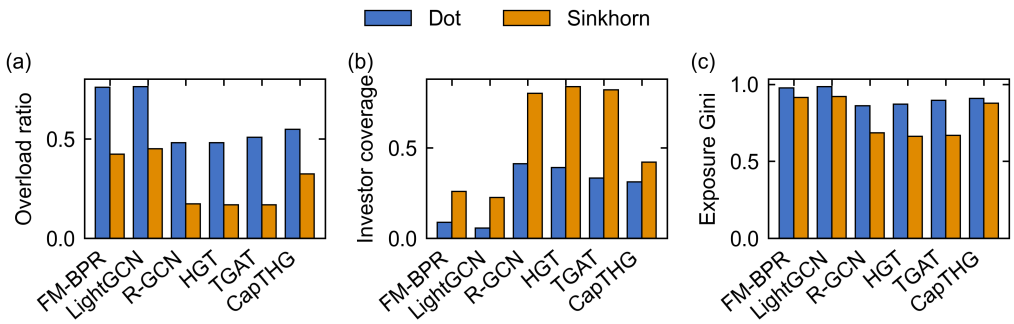


Figure 3. Capacity and exposure metrics of each learned encoder under the two ranking rules (Dot: score ranking; Sinkhorn: capacity-constrained Sinkhorn ranking). (**a**) Overload ratio OR@10; (**b**) investor coverage Cov@10; (**c**) exposure Gini coefficient Gini@10.

After 50 Sinkhorn iterations, the row and column constraints of the plan hold to within 0.6% relative error for every encoder, and for CapTHG they hold to the printed precision already after 20 iterations. The top-K lists, however, are a rounding of the continuous plan, so overload can remain in them, as the OR@10 of 0.3246 for CapTHG shows. Capacity-feasible lists are built from the same plan by taking pairs in decreasing plan value and placing investor i in at most q_i lists. The integer quotas q_i apportion the $|\mathcal{R}_q| \cdot K$ list positions of the quarter in proportion to b_i by largest-remainder rounding. Table 6 compares the two kinds of lists.

Table 6. Top-K lists read off the Sinkhorn plan (Plan top-K) and capacity-feasible lists built from the same plan by greedy assignment under integer quotas (Feasible top-K), mean over five seeds.

Encoder	Recall@20 (Plan)	Recall@20 (Feasible)	OR@10 (Plan)	OR@10 (Feasible)	Cov@10 (Feasible)
FM-BPR	0.1450	0.1234	0.4231	0.0378	0.9454
LightGCN	0.1523	0.1364	0.4509	0.0390	0.9454
R-GCN	0.2850	0.2962	0.1730	0.0379	0.9454
HGT	0.2713	0.2834	0.1689	0.0379	0.9454
TGAT	0.2414	0.2590	0.1689	0.0378	0.9454
CapTHG	0.3919	0.3607	0.3246	0.0378	0.9454

With capacity-feasible lists, OR@10 drops to 0.0378, which is exactly the overload created by rounding the proportional shares to integers, and Cov@10 reaches 0.9454. The Recall@20 of CapTHG falls from 0.3919 to 0.3607 under these lists. This value stays above every baseline under score ranking, the best of which is R-GCN at 0.3216, and above every baseline under capacity-feasible lists (R-GCN 0.2962, HGT 0.2834, TGAT 0.2590, LightGCN 0.1364, FM-BPR 0.1234). Capacity can thus be enforced on the final lists up to integer rounding without losing the accuracy lead of CapTHG. Section 5.5 examines how these results depend on the hyperparameters of CapTHG.

5.5. Sensitivity Analysis

This subsection reports four robustness checks of CapTHG, covering the hyperparameters of the model, the definition of the investment capacity b_i, the training mini-batch size and the candidate threshold. Each check changes one setting at a time and keeps the others at the reference setting ($\lambda = 0.5$, $\varepsilon = 0.1$, $d = 64$, $L = 2$, four-quarter capacity, mini-batches of 768 rounds, at least three deals before 2014). All checks are run over several seeds and form an exploratory robustness analysis rather than a search for the best setting. Over seeds 0–2, the reference setting reaches a Recall@20 of 0.3902 ± 0.0073 and an OR@10 of 0.3332 ± 0.0164. Over five seeds, its Recall@20 is 0.3919 ± 0.0060, against 0.3216 for the strongest baseline R-GCN. All values below are means $\pm$ standard deviations over seeds.

The first check varies the loss weight λ, the regularization strength ε, the embedding dimension d and the number of layers L, each over three seeds (Figure 4a–c). The loss weight λ has the largest effect. With $\lambda = 0$, which is the same setting as the w/o OT loss ablation in Section 5.3, Recall@20 is 0.2679 ± 0.0097 and OR@10 is 0.1684 ± 0.0019. At $\lambda = 0.1$, Recall@20 rises to 0.3570 ± 0.0076, so most of the gain from the plan likelihood loss occurs between $\lambda = 0$ and $\lambda = 0.1$. For $\lambda = 0.5$, 1.0 and 2.0, Recall@20 is 0.3902 ± 0.0073, 0.3931 ± 0.0037 and 0.3984 ± 0.0051, and it varies within 0.0082. OR@10 is 0.2894 ± 0.0640, 0.3332 ± 0.0164, 0.2999 ± 0.0240 and 0.3032 ± 0.0334 for $\lambda = 0.1$, 0.5, 1.0 and 2.0. Its seed-to-seed spread is of the same order as the differences between these weights.

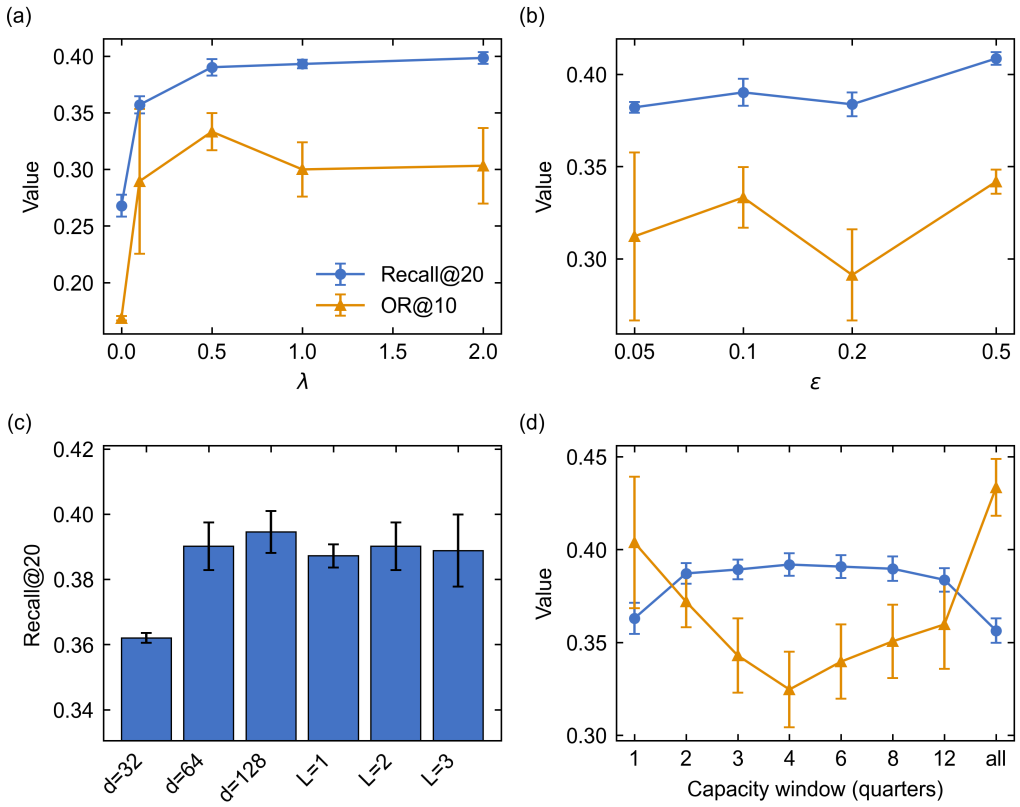


Figure 4. Sensitivity of CapTHG (mean ± standard deviation over seeds). (**a**) Loss weight λ and (**b**) regularization strength ε, Recall@20 and OR@10, three seeds; (**c**) embedding dimension d and number of layers L, Recall@20, three seeds; (**d**) length of the capacity window, Recall@20 and OR@10 scored against the four-quarter capacity, five seeds.

Regarding the regularization strength, the effect of ε is milder. For $\varepsilon = 0.05, 0.1, 0.2$ and 0.5, Recall@20 is 0.3821 ± 0.0029, 0.3902 ± 0.0073, 0.3837 ± 0.0064 and 0.4086 ± 0.0034. The corresponding OR@10 values are 0.3122 ± 0.0455, 0.3332 ± 0.0164, 0.2913 ± 0.0247 and 0.3418 ± 0.0065. All four strengths keep Recall@20 well above that of R-GCN, so the entropic regularization of the capacity-constrained Sinkhorn layer does not need fine adjustment.

For the encoder, an embedding dimension of 32, 64 and 128 gives Recall@20 of 0.3620 ± 0.0015, 0.3902 ± 0.0073 and 0.3946 ± 0.0065. A smaller embedding therefore costs accuracy, while a larger one adds less than one standard deviation. One, two and three layers give 0.3872 ± 0.0036, 0.3902 ± 0.0073 and 0.3888 ± 0.0111, and these differences also lie within one standard deviation.

Some settings give a slightly higher test Recall@20 than the reference setting, for example $\varepsilon = 0.5$ (0.4086), $\lambda = 2.0$ (0.3984) and $d = 128$ (0.3946). The reference setting was fixed before these checks and is kept throughout the paper. The main results are therefore not tuned to the test period. Every setting with $\lambda \geq 0.1$ stays above R-GCN.

The second check concerns the definition of the investment capacity b_i (Table 7). The window length of the activity average runs from 1 to 12 quarters and also covers all history, each with five seeds. The floor of the four-quarter average takes 0.1, 0.5 and 1.0, and an exponentially weighted activity with a half-life of two quarters is added, each with three seeds. OR@10 is always scored against the four-quarter capacity of Equation (6), so all definitions share one yardstick.

Table 7. CapTHG under alternative definitions of the investment capacity b_i (window length of the activity average with floor 0.25; floor of the four-quarter average; exponentially weighted activity). Mean ± standard deviation over seeds; OR@10 is scored against the four-quarter capacity of Equation (6) so that all rows share one yardstick.

Capacity Definition	Seeds	Recall@20	NDCG@10	OR@10
1 quarter	5	0.3630 ± 0.0084	0.2271 ± 0.0077	0.4038 ± 0.0354
2 quarters	5	0.3872 ± 0.0056	0.2468 ± 0.0060	0.3720 ± 0.0138
3 quarters	5	0.3893 ± 0.0053	0.2489 ± 0.0054	0.3430 ± 0.0200
4 quarters (main)	5	0.3919 ± 0.0060	0.2520 ± 0.0053	0.3246 ± 0.0203
6 quarters	5	0.3908 ± 0.0062	0.2538 ± 0.0053	0.3397 ± 0.0200
8 quarters	5	0.3897 ± 0.0066	0.2528 ± 0.0056	0.3506 ± 0.0197
12 quarters	5	0.3837 ± 0.0063	0.2500 ± 0.0066	0.3597 ± 0.0239
all history	5	0.3564 ± 0.0066	0.2313 ± 0.0066	0.4335 ± 0.0153
floor 0.1	3	0.3914 ± 0.0058	0.2491 ± 0.0051	0.3408 ± 0.0144
floor 0.5	3	0.3832 ± 0.0085	0.2474 ± 0.0064	0.3228 ± 0.0109
floor 1.0	3	0.3634 ± 0.0064	0.2351 ± 0.0060	0.3356 ± 0.0028
exp. weighted, half-life 2	3	0.3921 ± 0.0056	0.2521 ± 0.0052	0.3402 ± 0.0168

Windows of 2 to 12 quarters give Recall@20 between 0.3837 and 0.3919 (Figure 4d). OR@10 is lowest for the four-quarter window (0.3246) and rises for both shorter and longer windows, reaching 0.3720 at two quarters and 0.3597 at twelve. A one-quarter window (0.3630 ± 0.0084, OR@10 0.4038) and the all-history average (0.3564 ± 0.0066, OR@10 0.4335) are clearly worse. One quarter of activity is too noisy to measure capacity, and the all-history average lags behind current activity.

The floor has a weaker effect at small values. Floors of 0.1, 0.5 and 1.0 give Recall@20 of 0.3914 ± 0.0058, 0.3832 ± 0.0085 and 0.3634 ± 0.0064. A floor of one deal per quarter gives inactive investors the same capacity as active ones and flattens the capacity profile. The exponentially weighted activity reaches 0.3921 ± 0.0056 with an OR@10 of 0.3402, close to the four-quarter window. Every capacity definition keeps CapTHG above R-GCN.

A further check changes the definition only at inference. The five CapTHG models trained with the four-quarter capacity are evaluated with the other definitions in the capacity-constrained Sinkhorn ranking. For windows of 2–12 quarters, floors of 0.1 and 0.5 and the exponentially weighted measure, Recall@20 stays between 0.3841 and 0.3941. The learned encoder is thus not tied to the capacity definition used in training.

The third check varies the training mini-batch size over three seeds. Mini-batches of 384, 768 and 1536 rounds give Recall@20 of 0.3939 ± 0.0095, 0.3902 ± 0.0073 and 0.3900 ± 0.0093, and training on whole quarters gives 0.3902 ± 0.0096. These differences are below one standard deviation. To compare the batch plan with the full-quarter plan directly, 20 random batches of 768 rounds are drawn per test quarter for each of the five trained models, with capacities rescaled as in training. The top-10 lists of the two plans share an average Jaccard overlap of 0.846 ± 0.011, and their rows differ by a total variation of 0.058. On the same rounds, Recall@20 is 0.3836 from batch plans and 0.3852 from the full-quarter plan. The mini-batch approximation used in training therefore stays close to the plan over a whole quarter.

The fourth check lowers the candidate threshold from three deals before 2014 to two and to one (Table 8). Each threshold defines its own candidate investors and test rounds. With at least two deals (7240 candidates, 10,124 test rounds), CapTHG reaches 0.3512 ± 0.0080 against 0.2874 ± 0.0142 for R-GCN. With at least one deal (18,454 candidates, 10,842 test rounds), the values are 0.3040 ± 0.0120 and 0.2267 ± 0.0082. For both thresholds, CapTHG is higher in all 18 quarter-by-seed cells (one-sided Wilcoxon signed-rank test, $p = 3.8 \times 10^{-6}$) and in all 6 quarters (exact sign test, $p = 0.016$). Recall@20 of every model falls as the

candidate set grows, because the same rounds are ranked against more investors with short histories.

Table 8. Recall@20 under three candidate thresholds (deals before 2014). Each threshold defines its own candidate set and test rounds; the main setting uses five seeds, the relaxed thresholds three seeds (mean ± standard deviation); Popularity is deterministic.

Threshold	Candidates	Test Rounds	CapTHG	R-GCN	Popularity
≥3 (main)	4763	9679	0.3919 ± 0.0060	0.3216 ± 0.0071	0.1339
≥2	7240	10,124	0.3512 ± 0.0080	0.2874 ± 0.0142	0.1255
≥1	18,454	10,842	0.3040 ± 0.0120	0.2267 ± 0.0082	0.1135

In all four checks, CapTHG stays above the strongest baseline, so the conclusions of the preceding comparisons do not depend on these settings. Section 5.6 turns to the numerical verification of the theory.

5.6. Numerical Verification of the Theory

The four theoretical results are checked on five randomly generated instances. Each instance has a 12×20 score matrix, and the dustbin score is set to $\alpha = 0.3$. Table 9 summarizes the checks, and Figure 5 shows the corresponding curves.

Table 9. Controlled numerical checks of Proposition 1, Theorem 1, Theorem 2 and Proposition 2 on five random instances (12×20 scores, dustbin score $\alpha = 0.3$).

Check	Result
Real block satisfies $P\mathbf{1} \le a$, $P^\top \mathbf{1} \le b$ (5/5 instances)	yes
Max. error of the objective identity	7.1×10^{-15}
Max. LP gap at $\varepsilon = 0.01$	8.91×10^{-3}
LP gap within the bound of Theorem 1 (all ε, all instances)	yes
Empirical contraction rate, $\varepsilon = 0.5$	0.316
Empirical contraction rate, $\varepsilon = 0.2$	0.785
Empirical contraction rate, $\varepsilon = 0.1$	0.901
Max. $\lvert \partial W_\varepsilon / \partial S_{ri} - P^\star_{ri} \rvert$	4.8×10^{-10}

Proposition 1 holds on every instance. For each of the five instances, the real block of the transport plan lies in the feasible set of the inequality capacity constraints, so the feasibility check is passed in 5/5 cases. The maximum error between the two sides of the objective identity is 7.1×10^{-15}, which is of the order of floating-point precision. This result indicates that the dustbin enters the objective value as a retention threshold.

Theorem 1 is confirmed with respect to both the linear programming limit and the error bound. At $\varepsilon = 0.01$, the maximum gap between the entropic optimal value and the linear programming optimal value is 8.91×10^{-3}. Across all values of ε and all instances, the measured gap stays within the error bound $\varepsilon M \log N$. The observed gaps are therefore consistent with the bound stated in the theorem.

The convergence behavior predicted by Theorem 2 is reflected in the empirical contraction rates. The marginal error of the Sinkhorn iteration decreases at a geometric rate. The empirical contraction rates are 0.316, 0.785 and 0.901 at $\varepsilon = 0.5$, 0.2 and 0.1, respectively. These rates increase as ε decreases. This trend agrees with the contraction coefficient κ, according to which a smaller ε leads to slower convergence.

Proposition 2 is also verified. Its envelope identity relates the gradient of the entropic optimal value with respect to the score matrix to the optimal transport plan. The maximum deviation between this gradient and the obtained optimal transport plan is 4.8×10^{-10}.

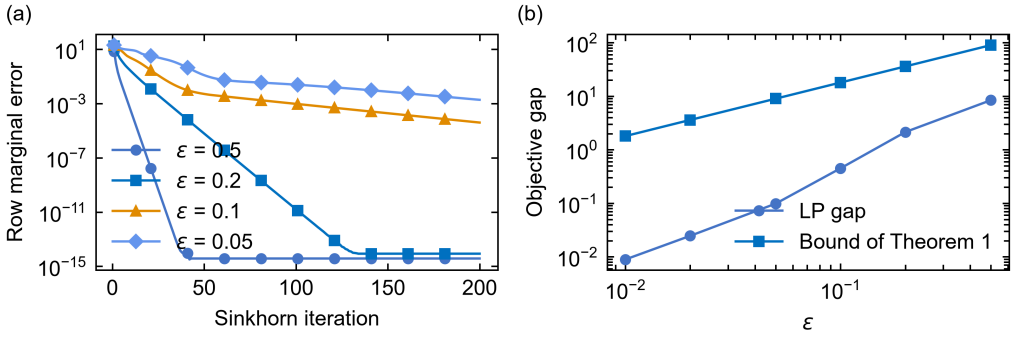


Figure 5. Marginal error per Sinkhorn iteration and the gap to the linear program. (**a**) Row marginal error versus Sinkhorn iteration for $\varepsilon = 0.5, 0.2, 0.1$ and 0.05; (**b**) gap between the entropic and the linear programming optimal values (LP gap) and the error bound of Theorem 1, as functions of ε.

5.7. Runtime

We measured the training time and inference time on a single CPU and averaged them over five seeds. The test period covers six quarters with 9679 financing rounds in total. CapTHG trains in 996.6 s, and early stopping is triggered at 17.6 epochs on average. Inference over all six quarters takes 9.8 s.

CapTHG has a lower training cost than the baseline models. R-GCN, HGT, LightGCN and TGAT require 1739.5 s, 2087.0 s, 2112.6 s and 4674.7 s of training, respectively. At inference, score ranking with the same encoder takes only 2.7 s. The additional inference time comes from the capacity-constrained Sinkhorn layer. This overhead is consistent with the $O(T|\mathcal{R}_q||\mathcal{I}|)$ complexity derived in Section 4.6.

6. Discussion

6.1. Why the Capacity Constraint Helps

The capacity constraint redistributes exposure inside the matching step rather than after ranking. The transport plan of the Sinkhorn layer must satisfy the row constraint and the column constraint at the same time. The column marginal of each investor is its capacity share, that is, the number of financing rounds it can absorb in the quarter. An investor with a high matching score whose share is already filled by better-suited rounds gives up its positions in the remaining rounds. Investors with spare capacity take over these positions in the order of their scores. The results of Section 5.4 agree with this mechanism. Under the same encoder, moving from score ranking to the capacity-constrained Sinkhorn ranking lowers the capacity overload ratio OR@10 from 0.5486 to 0.3246, raises the investor coverage Cov@10 from 0.3121 to 0.4204, and lowers the exposure Gini coefficient Gini@10 from 0.9086 to 0.8772. Overload falls and coverage rises for all six encoders, so the shift of exposure from saturated head investors to investors with spare capacity does not depend on the encoder.

Accuracy behaves differently. Recall@20 rises only for CapTHG, from 0.3452 to 0.3919, and falls for the five encoders trained with the pairwise ranking loss alone. With this gain, CapTHG exceeds the strongest baseline R-GCN, whose Recall@20 is 0.3216 (Section 5.2). Real financing rounds distribute their shares over several investors. Head investors can join only a limited number of rounds per quarter, and the remaining syndicate seats go to moderately active investors. Score ranking treats each round independently and therefore pushes the same head investors to the top of every round. The capacity-constrained plan returns those positions to the investors that actually join. This return produces hits only when the scores anticipate the column rescaling, and the plan likelihood loss teaches the encoder exactly this because it trains the scores through the plan. The same loss keeps the plan close to the observed syndication pattern. In the test quarters, the observed investment

edges have an overload ratio of 0.424 against the same capacity shares and involve 27.8% of the candidates per quarter. The plan of CapTHG, with OR@10 of 0.3246 and Cov@10 of 0.4204, stays close to this pattern, whereas encoders trained with the pairwise ranking loss spread their lists over 80–84% of the candidates under the Sinkhorn ranking.

The dustbin form extends the same mechanism to capacities that are upper bounds. Proposition 1 shows that, when the column marginals are relaxed from equalities to upper bounds, the augmented problem with a dustbin remains feasible and its solution keeps the capacity bound on the real block. In the dustbin variant, the learned dustbin score α is -0.20 on the standardized score scale, with values of -0.20, -0.17 and -0.24 for seeds 0–2, and its Recall@20 is 0.3688 (Section 5.3). The numerical verification in Section 5.6 gives convergence behavior consistent with this result.

6.2. Boundary of the Results

The investment capacity b_i is an activity-based proxy estimated from deal counts because fund size, investable capital and post-investment staff are not in the data. Within the tested definitions, the conclusions are stable (Section 5.5). Windows of 2–12 quarters give Recall@20 between 0.3837 and 0.3919, floors of 0.1 and 0.5 give 0.3914 and 0.3832, and an exponentially weighted measure gives 0.3921. Even the weakest definitions, namely a one-quarter window (0.3630), the all-history count (0.3564) and a floor of 1.0 (0.3634), stay above R-GCN at 0.3216. Financial capacity itself has not been tested, and the next step is to bring fund-level information into the capacity estimate.

The data are the seed, angel and venture rounds of the public Crunchbase snapshot for 2005–2015, with 4763 candidate investors and 9679 financing rounds in six test quarters. Candidate investors must have a deal history of at least three deals before 2014, and they take part in 59.1% of the investment edges of the test period. Investors with no deal before 2014 account for 28.8% of the edges and lie outside the scope of the model, since the representation of an investor is a learned embedding. Relaxing the threshold to one deal gives 18,454 candidates and keeps the advantage of CapTHG over R-GCN, with Recall@20 of 0.3040 against 0.2267, while accuracy falls as the candidate set grows (Section 5.5). The next steps are an inductive investor representation for new investors and a repetition of the rolling evaluation over a longer period and more round types.

The main model uses the equality-normalized form, in which capacity is normalized to an equality. Capacities that are upper bounds are covered by the dustbin form, whose Recall@20 on these data is 0.3688. The next step is to use the dustbin form directly in market segments where capacity is generally not exhausted.

The Sinkhorn layer is solved per quarter, so all rounds of a quarter enter the allocation at once and their order within the quarter is not modeled. The top-K lists are a rounding of the transport plan, and exactly capacity-feasible lists lower Recall@20 from 0.3919 to 0.3607 (Section 5.4). The next step is to write the within-quarter order and the integrality of the lists into the transport problem. Within these boundaries, the capacity-constrained transport plan improves both the accuracy of matching and the distribution of exposure across investors, which is the basis of the conclusions that follow.

7. Conclusions

This paper formulates the quarterly matching of startups and investors as an entropic optimal transport problem with capacity constraints and solves it with CapTHG, a temporal heterogeneous graph matching model.

For RQ1, the limit on the number of financing rounds each investor can join in a quarter and the limit on the number of investors each financing round can admit can be written as a differentiable matching layer. Proposition 1, Theorem 1, Theorem 2 and Proposition 2

give the feasibility, limit behavior, convergence and gradient of this layer, and the numerical verification confirms them. For RQ2, the relation-specific heterogeneous attention encoder on causal quarterly snapshots is trained jointly with the capacity-constrained Sinkhorn layer. In the quarterly rolling evaluation of real financing rounds, CapTHG recommends investors more accurately than comparable graph recommendation models, and its round-level Recall@20 is higher than that of the strongest baseline. For RQ3, the capacity-constrained Sinkhorn ranking lowers the capacity overload ratio, raises investor coverage and lowers the exposure Gini coefficient for every encoder. Accuracy rises together with these gains only for CapTHG, whose encoder is trained jointly with the transport plan. Capacity estimated from the four-quarter activity window works better than the all-history average, although capacity remains an activity-based proxy. Inference over a test period takes 9.8 s, against 2.7 s for score ranking.

The formulation is not tied to the financing market. Any two-sided matching in which one side can accept only a limited number of objects within a time window could be written as a transport problem with the same capacity constraint. Such a problem could then be solved with the same differentiable Sinkhorn layer. Quantitative modeling of how system-level behavior depends on interacting components and constraints is equally active in other fields, for example in antenna design and decoupling [68–70], the thermal conductivity of concrete [71], the mechanics of tunnel construction [72], nanoparticle-based immunotherapy [73], energy decentralization [74] and urban energy intensity [75], destination marketing [76], active battery equalization [77], battery pack voltage self-balancing [78], piezoelectric energy harvesting [79], lower-limb synergy modeling [80], strain field customization in bioreactors [81], cerebral blood flow prediction [82], heart sound representation learning [83], credit card risk identification [84], corroded pipeline maintenance optimization [85], traffic vehicle detection from remote sensing [86], steel bridge deck damage detection [87], and viscoelastic property prediction of asphalt mixtures [88].

Author Contributions: Conceptualization, B.Y. and C.H.; methodology, B.Y. and C.H.; software, B.Y.; validation, C.H.; formal analysis, B.Y.; investigation, B.Y.; data curation, B.Y.; writing—original draft preparation, B.Y.; writing—review and editing, C.H.; visualization, B.Y.; supervision, C.H.; project administration, C.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The study uses a publicly available Crunchbase snapshot. The raw file `investments.csv` was obtained from the public mirror https://github.com/notpeter/crunchbase-data (accessed on 19 September 2026), which redistributes the Crunchbase 2013–2015 export under the terms stated in that repository. The code that builds the temporal heterogeneous graph, trains every model reported here and regenerates all tables and figures, together with the derived result files, is provided in the replication package accompanying this article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Gompers, P.A.; Gornall, W.; Kaplan, S.N.; Strebulaev, I.A. How do venture capitalists make decisions? *J. Financ. Econ.* **2020**, *135*, 169–190. [CrossRef]
2. Sørensen, M. How Smart Is Smart Money? A Two-Sided Matching Model of Venture Capital. *J. Financ.* **2007**, *62*, 2725–2762. [CrossRef]
3. Yang, S.; Ren, X.; Zha, B.; Bao, Q.; Feng, B.; Xu, Z. EFSRNet: Multi-scale exposure normalization and dual-branch aggregation for overexposed face super-resolution. *Knowl.-Based Syst.* **2026**, *339*, 115578. [CrossRef]

4. Huang, J.; Huang, Z.; Xie, C.; Zhang, Y.; Ostachowicz, W.; Cao, M. Unsupervised deep learning framework for damage identification under ambient excitations: Trait of damage localization and demonstrative applications. *Mech. Syst. Signal Process.* **2026**, *250*, 114123. [CrossRef]
5. Yan, W.; Ren, J.; Feng, J.; Duan, Y.; Wei, C. A New Forest Fire Risk Rating Forecast Model Based on XGBoost. In Proceedings of the 2022 International Seminar on Computer Science and Engineering Technology (SCSET), Indianapolis, IN, USA, 8–9 January 2022; pp. 227–230. [CrossRef]
6. Wei, C.; Guo, C.; Yan, W. Forest Fire Risk Forecast Method with Pseudo Label Based on Semi-supervised Learning. In Proceedings of the 2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), Taiyuan, China, 3–5 December 2021; pp. 36–39. [CrossRef]
7. Zhou, X.; Li, Z.; Liu, X.; Wang, Y.; Liang, N.; Zhong, Z. Physics-informed neural network (PINN) for predicting the temperature field in high geothermal tunnels. *Tunn. Undergr. Space Technol.* **2026**, *169*, 107265. [CrossRef]
8. Wu, S.; Sun, F.; Zhang, W.; Xie, X.; Cui, B. Graph Neural Networks in Recommender Systems: A Survey. *ACM Comput. Surv.* **2023**, *55*, 1–37. . [CrossRef]
9. Hu, Z.; Dong, Y.; Wang, K.; Sun, Y. Heterogeneous Graph Transformer. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 2704–2710. [CrossRef]
10. Cuturi, M. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–8 December 2013; Volume 26.
11. Sarlin, P.E.; DeTone, D.; Malisiewicz, T.; Rabinovich, A. SuperGlue: Learning Feature Matching With Graph Neural Networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 4937–4946. [CrossRef]
12. Christakopoulou, K.; Kawale, J.; Banerjee, A. Recommendation with Capacity Constraints. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 1439–1448. [CrossRef]
13. Gao, C.; Zheng, Y.; Li, N.; Li, Y.; Qin, Y.; Piao, J.; Quan, Y.; Chang, J.; Jin, D.; He, X.; et al. A Survey of Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions. *ACM Trans. Recomm. Syst.* **2023**, *1*, 1–51.. [CrossRef]
14. He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; Wang, M. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, China, 25–30 July 2020; pp. 639–648. [CrossRef]
15. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; van den Berg, R.; Titov, I.; Welling, M. Modeling Relational Data with Graph Convolutional Networks. In *Lecture Notes in Computer Science*; Springer International Publishing: Cham, Switzerland, 2018; pp. 593–607. [CrossRef]
16. Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; Yu, P.S. Heterogeneous Graph Attention Network. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 2022–2032. [CrossRef]
17. Wang, X.; Bo, D.; Shi, C.; Fan, S.; Ye, Y.; Yu, P.S. A Survey on Heterogeneous Graph Embedding: Methods, Techniques, Applications and Sources. *IEEE Trans. Big Data* **2023**, *9*, 415–436. [CrossRef]
18. Wang, X.; Liu, N.; Han, H.; Shi, C. Self-supervised Heterogeneous Graph Neural Network with Co-contrastive Learning. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Virtual, 14–18 August 2021; pp. 1726–1736. [CrossRef]
19. Lv, Q.; Ding, M.; Liu, Q.; Chen, Y.; Feng, W.; He, S.; Zhou, C.; Jiang, J.; Dong, Y.; Tang, J. Are we really making much progress? Revisiting, benchmarking and refining heterogeneous graph neural networks. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Virtual, 14–18 August 2021; pp. 1150–1160. [CrossRef]
20. Li, S.; Jin, T.; Luo, H.; Wang, E.; Tao, R. Recommendation Model Based on Higher-Order Semantics and Node Attention in Heterogeneous Graph Neural Networks. *Mathematics* **2025**, *13*, 1479. [CrossRef]
21. Du, X.; Cao, J.; Jiang, X.; Duan, J.; Tian, Z.; Wang, X. Enterprise Bankruptcy Prediction Model Based on Heterogeneous Graph Neural Network for Fusing External Features and Internal Attributes. *Mathematics* **2025**, *13*, 2755. [CrossRef]
22. Xu, D.; Ruan, C.; Korpeoglu, E.; Kumar, S.; Achan, K. Inductive Representation Learning on Temporal Graphs. In Proceedings of the International Conference on Learning Representations (ICLR), Online, 26 April–1 May 2020.
23. Wang, Y.; Chang, Y.Y.; Liu, Y.; Leskovec, J.; Li, P. Inductive Representation Learning in Temporal Networks via Causal Anonymous Walks. In Proceedings of the International Conference on Learning Representations (ICLR), Online, 3–7 May 2021.
24. Fan, Y.; Ju, M.; Zhang, C.; Ye, Y. Heterogeneous Temporal Graph Neural Network. In Proceedings of the 2022 SIAM International Conference on Data Mining (SDM), Alexandria, VA, USA, 28–30 April 2022; pp. 657–665. [CrossRef]
25. Wang, Y.; Huang, T.; He, C.; Li, Q.; Gao, J. Simple and Efficient Heterogeneous Temporal Graph Neural Network. *Proc. Adv. Neural Inf. Process. Syst.* **2025**, *38*, 122331–122357. [CrossRef]
26. Cao, J.; Li, J.; Jiang, J. Link Prediction for Temporal Heterogeneous Networks Based on the Information Lifecycle. *Mathematics* **2023**, *11*, 3541. [CrossRef]

27. Cong, W.; Zhang, S.; Kang, J.; Yuan, B.; Wu, H.; Zhou, X.; Tong, H.; Mahdavi, M. Do We Really Need Complicated Model Architectures For Temporal Networks? In Proceedings of the International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
28. Yu, L.; Sun, L.; Du, B.; Lv, W. Towards Better Dynamic Graph Learning: New Architecture and Unified Library. *Proc. Adv. Neural Inf. Process. Syst.* **2023**, *36*, 67686–67700. [CrossRef]
29. Huang, S.; Poursafaei, F.; Danovitch, J.; Fey, M.; Hu, W.; Rossi, E.; Leskovec, J.; Bronstein, M.; Rabusseau, G.; Rabbany, R. Temporal Graph Benchmark for Machine Learning on Temporal Graphs. *Proc. Adv. Neural Inf. Process. Syst.* **2023**, *36*, 2056–2073. [CrossRef]
30. Zheng, Y.; Yi, L.; Wei, Z. A survey of dynamic graph neural networks. *Front. Comput. Sci.* **2025**, *19*, 196323. [CrossRef]
31. Nguyen Thi My, L.; Vo, T. An Integration of Fuzzy-Enhanced Heterophilic Graph Representation Learning for Improving Performance of Link Prediction. *Cybern. Syst.* **2026**, *57*, 2084–2112. [CrossRef]
32. Cui, Q.; Ding, Z.; Chen, F. Hybrid Directed Hypergraph Learning and Forecasting of Skeleton-Based Human Poses. *Cyborg Bionic Syst.* **2024**, *5*, 0093. [CrossRef] [PubMed]
33. Ye, W.; Wang, J.; Chen, L.; Dai, L.; Sun, Z.; Liang, Z. Adaptive Spatial–Temporal Aware Graph Learning for EEG-Based Emotion Recognition. *Cyborg Bionic Syst.* **2024**, *5*, 0088. [CrossRef] [PubMed]
34. Wang, H.; Wang, Y.; Yan, S.; Du, X.; Gao, Y.; Liu, H. Merge-and-Split Graph Convolutional Network for Skeleton-Based Interaction Recognition. *Cyborg Bionic Syst.* **2024**, *5*, 0102. [CrossRef] [PubMed]
35. Chen, Y.; Liu, Z.W.; Qin, Y. Finite-Time Topology Identification of Delayed Complex Dynamical Networks and Its Application. *Cyborg Bionic Syst.* **2024**, *5*, 0092. [CrossRef] [PubMed]
36. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian Personalized Ranking from Implicit Feedback. In Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI), Montreal, QC, Canada, 18–21 June 2009.
37. Sinkhorn, R.; Knopp, P. Concerning nonnegative matrices and doubly stochastic matrices. *Pac. J. Math.* **1967**, *21*, 343–348. [CrossRef]
38. Peyré, G.; Cuturi, M. Computational Optimal Transport with Applications to Data Sciences. *Found. Trends Mach. Learn.* **2019**, *11*, 355–607. [CrossRef]
39. Franklin, J.; Lorenz, J. On the scaling of multidimensional matrices. *Linear Algebra Its Appl.* **1989**, *114–115*, 717–735. [CrossRef]
40. Altschuler, J.; Niles-Weed, J.; Rigollet, P. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
41. Thibault, A.; Chizat, L.; Dossal, C.; Papadakis, N. Overrelaxed Sinkhorn–Knopp Algorithm for Regularized Optimal Transport. *Algorithms* **2021**, *14*, 143. [CrossRef]
42. Li, S.; Si, F.; Mo, D.; He, C. Stochastic Mirror-Prox Optimisation of Group Distributionally Robust Multimodal Logistic Regression for Multilingual Fact-Checking Verdict Prediction. *Mathematics* **2026**, *14*, 3574. [CrossRef]
43. Chizat, L.; Peyré, G.; Schmitzer, B.; Vialard, F.X. Scaling algorithms for unbalanced optimal transport problems. *Math. Comput.* **2018**, *87*, 2563–2609. [CrossRef]
44. Séjourné, T.; Peyré, G.; Vialard, F.X. Unbalanced Optimal Transport, from theory to numerics. In *Handbook of Numerical Analysis*; Elsevier: Amsterdam, The Netherlands, 2023; pp. 407–471. [CrossRef]
45. Yang, W.; Yang, J.; Liu, Y. Multimodal Optimal Transport Knowledge Distillation for Cross-domain Recommendation. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, Birmingham, UK, 21–25 October 2023; pp. 2959–2968. [CrossRef]
46. Chen, J.; Wan, J.; Ma, L. Regularized Discrete Optimal Transport for Class-Imbalanced Classifications. *Mathematics* **2024**, *12*, 524. [CrossRef]
47. Chen, S.; Lin, Y.; Zeng, Z.; Xue, M. Leveraging Attribute Interaction and Self-Training for Graph Alignment via Optimal Transport. *Mathematics* **2025**, *13*, 1971. [CrossRef]
48. Wang, X.; Guo, Q.; Wang, W. Graph-Regularized Survival Learning with Optimal Transport for Individual Employee Turnover Forecasting. *Mathematics* **2026**, *14*, 2747. [CrossRef]
49. Montesuma, E.F.; Mboula, F.M.N.; Souloumiac, A. Recent Advances in Optimal Transport for Machine Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 1161–1180. [CrossRef] [PubMed]
50. Żbikowski, K.; Antosiuk, P. A machine learning, bias-free approach for predicting business success using Crunchbase data. *Inf. Process. Manag.* **2021**, *58*, 102555. [CrossRef]
51. Ross, G.; Das, S.; Sciro, D.; Raza, H. CapitalVX: A machine learning model for startup selection and exit prediction. *J. Financ. Data Sci.* **2021**, *7*, 94–114. [CrossRef]
52. Kim, J.; Kim, H.; Geum, Y. How to succeed in the market? Predicting startup success using a machine learning approach. *Technol. Forecast. Soc. Change* **2023**, *193*, 122614. [CrossRef]
53. Maarouf, A.; Feuerriegel, S.; Pröllochs, N. A fused large language model for predicting startup success. *Eur. J. Oper. Res.* **2025**, *322*, 198–214. [CrossRef]

54. Chen, Y.; Si, F.; Lu, X.; Li, X. Research on the Influence Mechanism of the Across-Industrial-Chain Investment Speed on Innovation Performance of AI Enterprises: Improvement Path of Artificial Intelligence Technology Application. *Mob. Inf. Syst.* **2021**, *2021*, 6149746. [CrossRef]
55. Liang, Y.E.; Yuan, S.T.D. Predicting investor funding behavior using crunchbase social network features. *Internet Res.* **2016**, *26*, 74–100. [CrossRef]
56. Xu, S.; Zhang, Q.; Lü, L.; Mariani, M.S. Recommending investors for new startups by integrating network diffusion and investors' domain preference. *Inf. Sci.* **2020**, *515*, 103–115. [CrossRef]
57. Hwang, Y.; Jeon, E.; Balaci, A.T.; Hwang, J.; Lee, J. Who invests matters: A multimodal contrastive learning approach to successful startup-investor matching. *Inf. Process. Manag.* **2027**, *64*, 105072. [CrossRef]
58. Gale, D.; Shapley, L.S. College Admissions and the Stability of Marriage. *Am. Math. Mon.* **1962**, *69*, 9–15. [CrossRef]
59. Galichon, A.; Salanié, B. Cupid's Invisible Hand: Social Surplus and Identification in Matching Models. *Rev. Econ. Stud.* **2022**, *89*, 2600–2629. [CrossRef]
60. Palomares, I.; Porcel, C.; Pizzato, L.; Guy, I.; Herrera-Viedma, E. Reciprocal Recommender Systems: Analysis of state-of-art literature, challenges and opportunities towards social recommendation. *Inf. Fusion* **2021**, *69*, 103–127. [CrossRef]
61. Su, Y.; Bayoumi, M.; Joachims, T. Optimizing Rankings for Recommendation in Matching Markets. In Proceedings of the ACM Web Conference 2022, Lyon, France, 25–29 April 2022; pp. 328–338. [CrossRef]
62. Tomita, Y.; Togashi, R.; Hashizume, Y.; Ohsaka, N. Fast and Examination-agnostic Reciprocal Recommendation in Matching Markets. In Proceedings of the 17th ACM Conference on Recommender Systems, Singapore, 18–22 September 2023; pp. 12–23. [CrossRef]
63. Tomita, Y.; Yokoyama, T. Fair Reciprocal Recommendation in Matching Markets. In Proceedings of the 18th ACM Conference on Recommender Systems, Bari, Italy, 14–18 October 2024; pp. 209–218. [CrossRef]
64. Chen, J.; Dong, H.; Wang, X.; Feng, F.; Wang, M.; He, X. Bias and Debias in Recommender System: A Survey and Future Directions. *ACM Trans. Inf. Syst.* **2023**, *41*, 1–39. [CrossRef]
65. Abdollahpouri, H.; Mansoury, M.; Burke, R.; Mobasher, B.; Malthouse, E. User-centered Evaluation of Popularity Bias in Recommender Systems. In Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization, Utrecht, The Netherlands, 21–25 June 2021; pp. 119–129. [CrossRef]
66. Qi, Z.; Chen, Z.; Si, F. Multi-Algorithm Optimization of Explanation Stability in Ensemble Learning for Human Capital Attrition Risk Prediction. *Mathematics* **2026**, *14*, 3283. [CrossRef]
67. Demšar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
68. Ji, Z.; Sun, G.; Wang, K.; Wong, H.; Yu, Z.; Li, Z.; Wei, C.; Liu, P. A Circularly Polarized Complementary Antenna with Substrate Integrated Coaxial Line Feed for X-Band Applications. *Electronics* **2024**, *13*, 785. [CrossRef]
69. Wei, C.; Wu, K.L. Array-antenna decoupling surfaces for quasi-yagi antenna arrays. In Proceedings of the 2017 IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting, San Diego, CA, USA, 9–14 July 2017; pp. 2103–2104. [CrossRef]
70. Wu, W.; Cheng, Y.; Chen, S.; Wei, C.; Wang, G. Decoupling and bandwidth equalization for MIMO antennas by using novel network design. *Microw. Opt. Technol. Lett.* **2022**, *64*, 1462–1467. [CrossRef]
71. Zhou, X.; Li, Z.; Liu, X.; Wang, Y.; Du, L. Calculation model for the thermal conductivity of concrete considering microstructural characteristics. *Case Stud. Therm. Eng.* **2026**, *79*, 107789. [CrossRef]
72. Liu, X.; Fang, L.; Zhou, X.; Xu, Q.; Cheng, K. Mechanical response characteristics and influencing factors analysis of non-parallel complex subway station-tunnel groups close proximity construction. *Tunn. Undergr. Space Technol.* **2026**, *171*, 107472. [CrossRef]
73. Xia, Y.; Shi, B.; Wang, K.; Hu, L.; Wang, Q.; Xu, S.; Wang, X.; Xu, P.; She, Y.; Xie, H.; et al. A trinity STING-activating nanoparticle harnesses cancer cell STING machinery for enhanced immunotherapy. *J. Control. Release* **2025**, *377*, 256–266. [CrossRef] [PubMed]
74. Mao, F.; Lin, J.; Hou, Y. Dynamic evolution and frontiers of energy decentralization research: A bibliometric-based review. *Energy Strategy Rev.* **2026**, *64*, 102097. [CrossRef]
75. Xiao, B.; Guo, X.; Si, F. Reassessing determinants of urban energy intensity in China: Insights from controllable and uncontrollable factors. *Environ. Sci. Pollut. Res.* **2024**, *31*, 38367–38384. [CrossRef] [PubMed]
76. Liu, Y.; Chen, Y.; Peng, K.; Si, F. Tourist perception of city marketing quality: Scale development and validation. *Asia Pac. J. Tour. Res.* **2025**, *30*, 1610–1640. [CrossRef]
77. Zhu, B.; Yan, J.; Hu, M.; Yang, Y.; Ji, B.; She, X. A Highly Simplified Active Battery Equalizer Based on a High-Accuracy Series-Port Converter With Voltage Self-Equalization. *IEEE Trans. Ind. Electron.* **2026**, *early access*. [CrossRef]
78. Zhu, B.; Liu, C.; She, X.; Wu, H.; Yang, Y. A Series-Port Voltage High-Precision Self-Balancing Based on Flyback Converter Circuit for Active Balancing of Battery Packs. *IEEE Trans. Power Electron.* **2026**, *early access*. [CrossRef]
79. K.S., J.; Visalakshi, S. Enhanced Open-Circuit Voltage Subject to Harmonic Excitations in Piezoelectric Energy Harvesters Using Hybrid Optimization Algorithm. *Cybern. Syst.* **2026**, *57*, 1060–1083. [CrossRef]

80. Liang, F.; Mo, L.; Sun, Y.; Guo, C.; Gao, F.; Liao, W.H.; Cao, J.; Li, B.; Song, Z.; Wang, D.; et al. Interlimb and Intralimb Synergy Modeling for Lower Limb Assistive Devices: Modeling Methods and Feature Selection. *Cyborg Bionic Syst*. **2024**, *5*, 0122. [CrossRef] [PubMed]
81. Wang, J.; Sarkar, D.; Mohan, A.; Lee, M.; Ma, Z.; Chortos, A. Deep Learning for Strain Field Customization in Bioreactor with Dielectric Elastomer Actuator Array. *Cyborg Bionic Syst*. **2024**, *5*, 0155. [CrossRef] [PubMed]
82. Cai, L.; Liu, Y.; Li, K.; Xing, C.; Xu, Z.; Zhao, L.; Lv, K.; Li, Z.; Wang, H.; Wang, L.; et al. Multimodal Imaging-Based Cerebral Blood Flow Prediction Model Development in Simulated Microgravity. *Cyborg Bionic Syst*. **2025**, *6*, 0448. [CrossRef] [PubMed]
83. Qian, K.; Bao, Z.; Zhao, Z.; Koike, T.; Dong, F.; Schmitt, M.; Dong, Q.; Shen, J.; Jiang, W.; Jiang, Y.; et al. Learning Representations from Heart Sound: A Comparative Study on Shallow and Deep Models. *Cyborg Bionic Syst*. **2024**, *5*, 0075. [CrossRef] [PubMed]
84. Zhang, Y.; Zhang, Y.; Liang, Y.; Mu, S.; Zhang, Z.; Chen, X. AD-VGF: An Improved Generative Adversarial Network for Credit Card Risk Identification and Management in Digital Economy. *Cybern. Syst*. **2026**, *57*, 2058–2083. [CrossRef]
85. Cui, B.; Wang, H. Optimization of risk-based and cost-minimized maintenance strategy for corroded pipeline using reinforcement learning. *Reliab. Eng. Syst. Saf*. **2026**, *276*, 112929. [CrossRef]
86. Liu, Z.; Kwan, M.P.; Jiang, W.; Liu, Y.; Cui, B. A lightweight multiscale fusion framework for traffic vehicle detection from satellite remote sensing, UAV, and CCTV imagery. *IEEE Trans. Geosci. Remote Sens*. **2026**, *64*, 5632717. [CrossRef]
87. Liu, Z.; Gu, X.; Kwan, M.P.; Cui, B. Integrated ultrasonic testing and numerical simulation for damage detection in steel bridge deck pavements. *Eng. Struct*. **2026**, *355*, 122421. [CrossRef]
88. Cui, B.; Wang, H. Coupled prediction of viscoelastic properties of asphalt mixtures using a multi-task physics-constrained neural network. *Adv. Eng. Inform*. **2026**, *72*, 104437. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.