# LLMs Meet VC Graphs: Path-Based Reasoning and Adaptive Fusion for Startup Success Prediction

Anonymous ACL submission

## Abstract

Most venture capital (VC) investments on startups fail, while a few yield outsized gains. Accurately predicting startup success is thus crucial. Graph-based models confirm the value of structural signals but offer limited reasoning, whereas large language models (LLMs) provide strong reasoning and broad knowledge yet hallucinate without domain grounding. A core challenge is therefore to align LLM reasoning with explicit multi-hop graph paths and fuse these paths with unstructured evidence. Classic retrieval-augmented generation (RAG) mitigates this via textual evidence but overlooks high-order investor-company relations. Embedding-based graph RAG encodes such relations while discarding the explicit chains LLMs exploit. We propose MIRAGE-VC, a multi-perspective RAG framework for VC prediction. Our approach couples semantic retrieval with an information-gain–guided, stepwise path retriever that selects a compact set of cross-typed paths as explicit evidence. Specialized agents analyze heterogeneous sources, and a learnable gate weights their signals before a final LLM decision. On a real-world VC dataset, MIRAGE-VC achieves state-of-the-art performance with a 5.0% relative F1 gain and a 16.6% relative Precision@5 gain over the best baseline. Our implementation is available.[1]

## 1 Introduction

In venture capital (VC), accurately identifying high-potential startups is crucial given its high-risk, high-reward nature: from 1985 to 2009, roughly 60% of VC-backed firms lost money, while only 10% returned over five times the initial investment (Kerr et al., 2014). Consequently, predicting startup success becomes an increasingly important task.

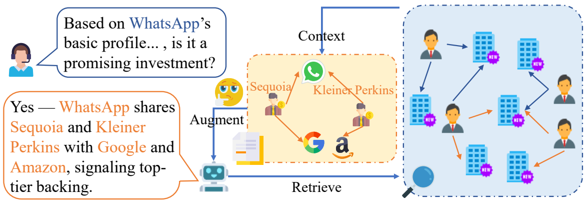
Based on WhatsApp's basic profile... , is it a promising investment?
Yes — WhatsApp shares Sequoia and Kleiner Perkins with Google and Amazon, signaling top-tier backing.
Context
Augment
Retrieve
Sequoia
Kleiner Perkins

Figure 1: Illustration of how the path retrieved from the graph by the path selector affects the prediction.

Early approaches relied on firm-level features and classic classifiers (e.g., logistic regression, random forests) (Sharchilev et al., 2018). More recent work leverages graph neural networks (GNNs) to capture complex dependencies in dynamic investment networks, demonstrating that structured graph representations further enhance VC prediction performance (Lyu et al., 2021; Zhang et al., 2021). However, GNNs rely solely on graph data and cannot perform explicit reasoning or integrate external knowledge. Recent advances in large language models (LLMs) have further expanded the toolkit for startup forecasting (Zhao et al., 2023). Unlike GNNs, which are limited to the observed graph structure, LLMs such as GPT (Achiam et al., 2023), with their strong reasoning capabilities and broad world knowledge, have been applied to improve both prediction accuracy and interpretability in this setting (Liu et al., 2023; Maarouf et al., 2025; Ko and Lee, 2024). However, LLMs are prone to hallucinations and often lack the domain grounding required for high-stakes, specialized tasks (Zhao et al., 2023). Retrieval-augmented generation (RAG) frameworks address this by integrating external knowledge without requiring fine-tuning (Lewis et al., 2020). Yet, classic RAG is optimized for unstructured textual data (Barnett et al.,

[1] https://anonymous.4open.science/r/MIRAGE-VC-C1E0

2024) and struggles to capture the high-order, multi-hop relationships common in VC investment graphs. As a result, its effectiveness in this domain remains limited.

Prior work extends RAG to graph-structured data by leveraging GNN-derived node embeddings (Mavromatis and Karypis, 2024). However, embedding-based retrieval discards explicit relational chains, which are essential for LLMs that excel at decomposing problems into interpretable, stepwise reasoning (Wei et al., 2022). In contrast, as shown in Figure 1, path-based retrieval enables extraction of semantically meaningful investment chains - e.g., “WhatsApp ← Sequoia → Google ← Kleiner Perkins → Amazon” which injects three actionable signals—investor quality via repeated early wins, cross-domain capability, and a reinforced coalition of top-tier backers-thus increasing the prior probability of success. These structured paths align well with LLMs’ chain-of-thought reasoning, facilitating more transparent and accurate predictions.

Despite its promise, integrating multi-hop relational paths with unstructured evidence sources remains challenging. First, graph-based reasoning is complex: shallow subgraphs around a target company may lack signals (Yu et al., 2021), yet as the hop count increases the number of candidate paths grows exponentially, and deeper expansions introduce noise through redundant or weakly informative paths (Zhang et al., 2025). Second, there is significant semantic heterogeneity across information sources: company documents offer business and market insights; investor profiles reflect experience and reputation, and graphs encode structural investment patterns. Naively merging these sources risks semantic conflicts and attention dilution within LLM prompts (Lv et al., 2024). Third, the relevance of each evidence type varies by startup category. For example, technology ventures often rely more on strong investor endorsements and network position (Sorenson and Stuart, 2001), while consumer-facing startups are better judged by operational indicators like market traction and growth rates (Belleflamme et al., 2014). Without a mechanism for dynamically weighting these perspectives, predictions risk over-reliance on secondary signals.

To address these challenges, we propose MIRAGE-VC, a multi-perspective retrieval-augmented framework for VC prediction, composed of three key components: **Path-Level Reasoning**: We introduce an information-gain-driven path retriever that iteratively selects a small set of high-order, cross-type investment chains from the VC graph, balancing informativeness and interpretability, to serve as explicit relational evidence. **Multi-Perspective Fusion**: We construct three evidence streams — company disclosures (Badertscher et al., 2013), lead investor profiles (Bernstein et al., 2022), and graph-based relational paths — each formatted as a structured, timestamped text blocks. Dedicated analysis agents reason over each stream independently before passing intermediate results to a central aggregation agent. **Adaptive Weighting**: A learnable gating network conditions on both the analysis outputs and the target company’s profile to compute normalized weights. These are injected into the final decision prompt, enabling the model to emphasize the most relevant evidence per prediction. The contributions are as follows.

- We propose the first RAG-based VC prediction framework that integrates unstructured document semantics with investment network graphs, enabling real-time multi-source knowledge injection and explicit chain-of-thought reasoning without fine-tuning the underlying LLM.

- We design a novel, information-gain-based path retriever and a multi-perspective fusion pipeline that transforms semantically heterogeneous company disclosures, lead investor profiles, and graph-based paths into structured prompts for dedicated LLM-based agents.

- Under strict measures to prevent data leakage into the LLM, our method achieves state-of-the-art performance on a real-world VC dataset—yielding relative improvements of 5.0% in F1 and 16.6% in Precision@5 over the best baseline.

## 2 Related Work

### 2.1 Graph-based VC Prediction

Traditional machine learning predictors rely on independent firm-level features and ignore relational context (Arroyo et al., 2019; Bento, 2017), whereas GNNs model investor–company graphs to capture high-order relational signals. SHGMNN (Zhang et al., 2021) combines predefined meta-paths, lightweight GNNs and Markov random field inference to integrate heterogeneous topologies and propagate labels for large-scale early-stage startup identification. GST (Lyu et al., 2021) applies unsupervised graph self-attention to update a dynamic startup-investor bipartite graph, improving node embeddings via link prediction and node classification to capture rich investor–company relations. These studies demonstrate that the structural properties of VC investment networks can significantly improve predictive accuracy. However, they remain limited by narrow knowledge scopes, weak reasoning capabilities, and a lack of interpretability.

### 2.2 Retrieval Augmented Generation

RAG has become a widely used strategy for grounding LLMs in external knowledge (Wiratunga et al., 2024; Jeong et al., 2024; Wang et al., 2025a). The standard RAG framework (Lewis et al., 2020) reduces hallucinations by retrieving relevant passages. However, it operates solely on linear, unstructured text and lacks mechanisms to capture structural dependencies or facilitate multi-hop reasoning over heterogeneous investment networks. Recent GNN-RAG frameworks (Mavromatis and Karypis, 2024) attempt to bridge this gap by retrieving contextually relevant nodes based on node embedding similarity. While this captures local structural cues, it still lacks explicit reasoning paths, preventing LLMs from fully leveraging their inherent strengths in explicit, stepwise, and interpretable chain-of-thought reasoning (Wei et al., 2022). As a result, these methods limit both the reasoning depth and interpretability that LLMs can offer in graph-based prediction tasks.

## 3 Preliminary

### 3.1 Problem definition

This study aims to predict the success of early-stage start-ups, defined as companies that have completed their first formal financing round (seed or angel) but have not yet raised Series A funding (Zhang et al., 2021). While success is often measured by the attainment of Series A financing (Zhang et al., 2021), prior studies use varying observation windows, which can introduce temporal bias. To mitigate this, we adopt a consistent one-year observation window following the seed round. This approach aligns with stage-based evaluation practices and helps control for external environmental factors (Boocock and Woods, 1997). The core task is to predict whether an early stage startup will secure subsequent financing within one year of its initial funding.

### 3.2 Data overview

We use the PitchBook[2] Global VC dataset, which spans investment activities from 2005 to November 2023. The dataset includes detailed investment records specifying the invested company, investor identity, funding amount, and financing stage. It also contains demographic information on both entrepreneurs and investors, including background, location, education, and professional biographies. Additionally, startup-level attributes are provided, such as team composition, industry classification, keyword tags, and geographic location. In total, the dataset encompasses 263,729 startups and 1,014,157 individuals.

### 3.3 VC investment network

We model the VC ecosystem as a time-stamped heterogeneous information network $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \mathcal{V}_{\text{cmp}} \cup \mathcal{V}_{\text{inv}}$ contains company nodes and investor nodes. Each directed edge $e = (v_{\text{inv}}, v_{\text{cmp}}, t)$ represents an investment event from investor to company at time $t$, annotated with attributes such as the financing round and investment amount. For each company $c^*$ that completes an angel or seed round at time $t$, we assign a binary label $y^* = 1$ if it secures Series A funding within the following 12 months, and $y^* = 0$ otherwise.

## 4 Methodology

### 4.1 Overview of our method

As shown in Figure 2, our proposed framework follows four sequential stages: (1) Graph retrieval: To supply the investment chain agent

[2] PitchBook is a financial data platform providing comprehensive information on private and public capital markets, including venture capital, private equity, and M&A transactions.

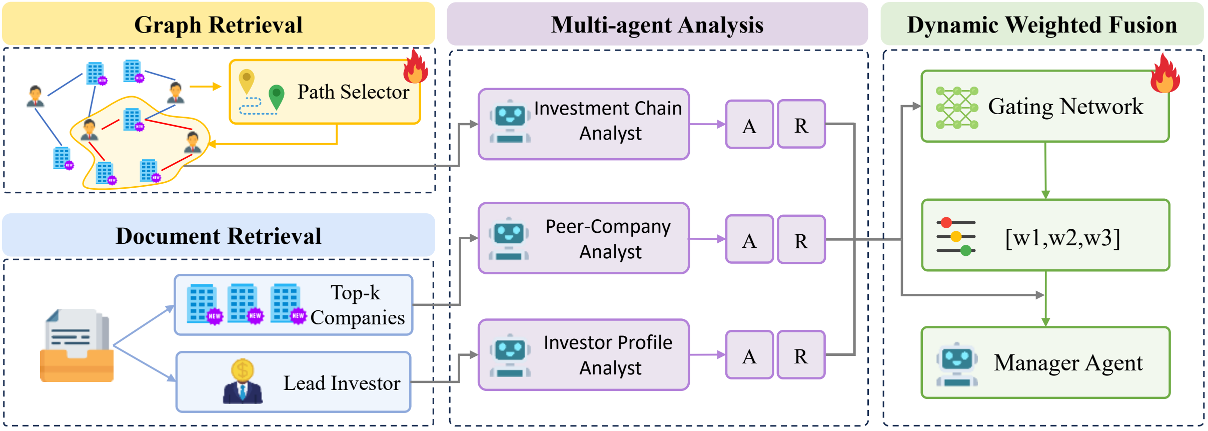

Figure 2: Overall Framework of MIRAGE-VC. It contains four key components: *Graph Retrieval* for path selecting, *Document Retrieval* for companies and investor, *Multi-agent Analysis* for multi-perspective information, *Dynamic Weighted Fusion* for adaptive information fusion.

with structured evidence, a learnable graph retriever extracts a high-value company–investor path from the investment graph. (2) Document retrieval: To provide the company and investor agents with textual context, semantic matching over public filings yields two textual views: (i) a similar company context and (ii) a lead-investor profile composed of demographics, career history, and labeled deal records. (3) Multi-agent analysis: The three prompts are processed by frozen LLM agents, each independently returning a binary decision and supporting rationale. (4) Perspective fusion: A lightweight gating network embeds and weighs the agent outputs. These are passed to a frozen manager agent, which produces a calibrated success probability and interpretable final decision.

## 4.2 Graph Retrieval

### 4.2.1 From classic IG to graph paths

As shown in Figure 3, path selection is framed as a sequential node selection problem. Starting from the target node $c^*$, at each hop, we choose the neighbor whose inclusion maximally improves the model's prediction accuracy. Although this heuristic does not guarantee a globally optimal path, it provides an efficient approximation—analogous to decision-tree splits via information gain—well suited to our multi-hop retrieval setting (Quinlan, 1986):

$$\mathrm{IG}(A) = H(Y) - H(Y \mid A) \tag{1}$$

We extend this principle to graphs by treating each candidate node $v$ as an "attribute" $A$ and estimating label uncertainty using the cross-entropy of a frozen LLM predictor. The rest of this section describes how these LLM-based IG signals are annotated offline, and how a lightweight selector model is trained to approximate them during inference.

### 4.2.2 LLM-generated gain labels

To obtain oracle supervision for our path selector, we use a frozen LLM to quantify the task-specific information gain of each candidate expansion. For each target company $c^*$, we build a breadth-first expansion tree of depth at most three, retaining up to three previously unseen neighbors per node. At hop $h \in \{0, 1, 2\}$ let $S^{(h)} = \langle c^*, \ldots, u\rangle$ denote a current path and $\{v_1, v_2, v_3\} \subseteq N(u)\backslash S^{(h)}$ the corresponding candidate set.

**Prompt construction** To measure the incremental value of each candidate node $v_i$, we generate two prompts per expansion: (i) a *baseline* prompt $P_{\text{base}}$ that verbalizes nodes in $S^{(h)}$, and (ii) a *candidate* prompt $P_{v_i}$ that verbalizes the extended path $S^{(h)}_{v_i} = \langle c^*, \ldots, u, v_i\rangle$. A frozen LLAMA-3.1-8B classifier returns the success probabilities $p_{\text{base}}$ and $p_{v_i}$. The procedure for converting causal-LM logits into binary probabilities $p$ is detailed in Appendix.

**Task-specific information gain** Given the gold label $y \in \{0, 1\}$ ($1 = $ *Success*, $0 = $ *Failure*), we define the marginal gain from including $v_i$ as:

$$\Delta_{v_i} = \underbrace{\mathrm{CE}(y,\, p_{\text{base}}) - \mathrm{CE}(y,\, p_{v_i})}_{\text{cross-entropy reduction}}$$
$$+ \lambda_{\text{conf}}\big(|p_{v_i} - 0.5| - |p_{\text{base}} - 0.5|\big) \quad (2)$$

where the first term rewards the reduction in the prediction error (irrespective of $y = 0$ or 1); the second encourages confidence once the correctness is taken into account. CE denotes binary cross-entropy and $\lambda_{\text{conf}} \in [0, 1]$ balances correctness against confidence shift.

**Training tuples** Each training instance consists of: $\big(S^{(h)},\ S^{(h)}_{v_i},\ \Delta_{v_i}\big)$ The selector later receives the baseline path $S^{(h)}$, the extended path $S^{(h)}_{v_i}$, and the scalar gain $\Delta_{v_i}$ it should learn to predict. Because gains are computed for both successful and failed companies, the selector is explicitly trained to prefer extensions that push the LLMs towards the correct class with higher confidence.

### 4.2.3 Selector training objective

Each hop $h$ of a target company contributes one *ranking group* $G^{(h)} = \{v_1, v_2, v_3\}$ with associated gains $\Delta_{v_1}, \Delta_{v_2}, \Delta_{v_3}$ annotated as in Eq. (2). For each candidate $v \in G^{(h)}$, we compute a difference feature:

$$x_v = \big[e_{\text{base}} \parallel e_v \parallel (e_v - e_{\text{base}})\big] \in \mathbb{R}^{2304} \quad (3)$$

where $e_{\text{base}}$ and $e_v$ are 768-dimensional sentence embeddings extracted once by a frozen encoder. A lightweight two-layer MLP $s_\theta : \mathbb{R}^{2304} \to \mathbb{R}$ assigns a score to each expansion.

**Listwise objective** To match the full gain pattern within each group we optimize a listwise objective. We first apply a within-group shift $r_i = \Delta_{v_i} - \min_j \Delta_{v_j}$, which preserves ordering while ensuring non-negativity ($r_i \geq 0$ and $\arg\max_i r_i = \arg\max_i \Delta_{v_i}$). We then form temperature-smoothed targets

$$q_i = \frac{\exp(r_i/\tau)}{\sum_{j=1}^{k} \exp(r_j/\tau)} \quad (4)$$

$$p_i = \frac{\exp(s_\theta(x_{v_i})/\tau)}{\sum_{j=1}^{k} \exp(s_\theta(x_{v_j})/\tau)} \quad (5)$$

The selector aligns its scores to the oracle distribution via

$$\mathcal{L}_{\text{list}}(G^{(h)}) = \mathrm{KL}(q \,\|\, p) = \sum_{i=1}^{k} q_i\big(\log q_i - \log p_i\big) \quad (6)$$

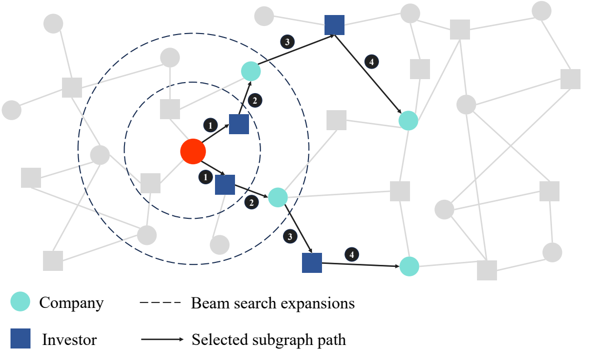

Figure 3: Illustration of how the path selector retrieves the best path from graph

where $\tau > 0$ controls target smoothness. Groups with $\sum_i r_i = 0$ carry no loss and are skipped. The final objective sums the listwise loss over all groups:

$$\mathcal{L}(\theta) = \sum_h \mathcal{L}_{\text{list}}(G^{(h)}) \quad (7)$$

This listwise training reproduces the within-group gain ranking and concentrates probability mass on high-gain candidates, enabling top-1 expansion at inference without re-invoking the LLM.

## 4.3 Document Retrieval

### 4.3.1 Company Retrieval

To place the target company with historically comparable cases, we retrieve companies whose public descriptions are semantically similar to that of the target. The intuition is that companies sharing industry focus, product form, or market stage provide informative priors on likely financing outcomes. We use a frozen sentence encoder to embed each description and rank candidates by cosine similarity (Mikolov et al., 2013)—retaining the top-$k$ peers $\mathcal{N}_k$. To avoid temporal leakage, we only consider firms founded before the target.

### 4.3.2 Investor Retrieval

Early capital often comes with intensive screening; hence the background of the lead investor provides strong priors about a startup future trajectory. We therefore identify, for the target company, the lead investor $v^*$ as the one who committed the largest amount in its first disclosed financing round at time $t_0$.

From PitchBook entries for $v^*$, we extract two types of time-stamped records and discard any entry with timestamp $t \geq t_0$ to avoid leakage: empirical records $H^{\text{emp}} = \{(r^{\text{emp}}_k, t^{\text{emp}}_k)\}$

and investment records $H^{\text{inv}} = \{(c_k^{\text{inv}}, t_k^{\text{inv}})\}$, where $r_k^{\text{emp}}$ denotes a role or title held by $v^*$ and $c_k^{\text{inv}}$ denotes a company previously backed by $v^*$. Both lists are ranked by recency and truncated to the top $n$ items.

For every invested company $c_k^{\text{inv}}$ we attach its brief profile and historical outcome label $\text{label}(c_k^{\text{inv}}) \in \{\textsc{Success}, \textsc{Failure}\}$. In addition, we collect static demographic attributes of $v^*$ (education, age, gender), denoted $A^*$. The resulting structured summary $\{A^*, H^{\text{emp}}, H^{\text{inv}}\}$ is verbalised into a investor-analysis prompt.

### 4.4 Multi-agent Analysis

We instantiate three specialist LLM agents, each mimicking a typical VC due-diligence role, to elicit complementary evidence. The **Peer-Company Analyst (PC)** agent examines the similar-company prompt built from peer-company documents. The **Investor Profile Analyst (IP)** agent reads an investor-analysis prompt that summaries the lead investor's biography and historical portfolio. And the **Investment Chain Analyst (IC)** agent reasons over a graph-path prompt that presents information-gain–optimized chains from the investment network. Each prompt is processed by the same frozen GPT-3.5 Turbo, which outputs a binary verdict and accompanying free-form rationale. Because the LLM backbone is shared and frozen, any differences in output reflect differences in evidence alone.

### 4.5 Gating Network

We formalize rationale fusion as a supervised weighting problem: each agent's textual rationale is jointly embedded with the target company's structured profile, and a lightweight gating network learns instance-specific weights to aggregate evidence for binary classification. This approach preserves richer evidence than scalar scores, yields interpretable weights linked to their supporting sentences, and maintains a frozen backbone LLM.

For each target company, we dispose of three rationales $R_i$ ($i \in \{\textbf{PC}, \textbf{IP}, \textbf{IC}\}$) produced by the specialist agents. Each rationale is embedded using a frozen sentence encoder: $r_i = f_{\text{enc}}(R_i) \in \mathbb{R}^{d_r}$. The company's structured attributes (industry, stage, region) are represented by a fixed vector $a^* \in \mathbb{R}^{d_a}$. A two-layer MLP $g_\phi$ scores every view conditioned on the instance.

$$s_i = g_\phi\big([\, r_i \,\|\, a^*]\big) \tag{8}$$

$$w_i = \frac{\exp(s_i)}{\sum_j \exp(s_j)} \tag{9}$$

The weights $w_i$ vary from case to case. The gated representation is the convex combination

$$r_{\text{f}} = \sum_i w_i \, r_i \;\in\; \mathbb{R}^{d_r} \tag{10}$$

which is concatenated with the attributes, is passed to another two-layer MLP $h_\theta$ followed by a sigmoid to obtain the success probability

$$p = \sigma\big(h_\theta([\, r_{\text{f}} \| a^*])\big) \;\in\; (0,1) \tag{11}$$

With ground-truth label $y \in \{0,1\}$ the gating parameters $\{\phi, \theta\}$ are learned by binary cross-entropy

$$\mathcal{L}(\phi, \theta) = -\, y \log p - (1-y) \log(1-p) \tag{12}$$

The softmax coefficients $\{w_i\}$ therefore offer an explicit, per-instance attribution of how much the company text, investor text, and graph path perspectives contribute to the final verdict.

### 4.6 Manager Agent

To obtain a comprehensive and human-readable final decision, the following artifacts are collated into a meta-prompt and forwarded to an additional frozen GPT-3.5 Turbo instance (the decision/manager agent): the target company's profile, the four agent predictions $\hat{y}_i$, the four rationales $R_i$, and the learned importance weights $w_i$. Conditioned on this structured input, the decision agent produces: (i) a final binary decision $\hat{y}_{\text{final}} \in \{True, False\}$ and (ii) a natural language explanation grounded in the individual rationales and their respective weights. This architecture provides interpretable, multi-source decision-making aligned with human VC analysis practices.

## 5 Experiments & Results

### 5.1 Datasets

**1. Dataset Sampling and Splitting** We train both the Path Selector and Gate Network on subsets drawn from the original investment graph which contains a large pool of candidate companies. To reduce computational overhead and avoid redundancy, we randomly sample 2,000 and 11,000 companies—respecting the overall success-to-failure ratio—for the Path Selector and Gate Network, respectively. Each subset is split into training, validation, and test sets in a $70 : 15 : 15$ ratio, with class balance maintained across splits.

**2. Final evaluation** We select 2,507 startups that successfully secured their first round of financing between October 2021 to November 2023—entirely after the LLM's pretraining cutoff—to prevent any test instances from appearing in the pretraining corpus and avoid data leakage. This set includes 1,974 negative samples and 533 positive samples.

## 5.2 Baselines

We compare our model with state-of-the-art baselines across four categories: GNN-based methods (SHGMNN, GST), embedding-based methods (BERT Fusion), RAG-based LLM methods (RAG, GNN-RAG), and recent LLM-driven VC predictors (SSFF).

**SHGMNN** (Zhang et al., 2021) aggregates the heterogeneous network by meta-paths into a graph and applies a diffusion GNN with convex MAP inference in a variational EM loop to model label dependencies. **GST** (Lyu et al., 2021) models the evolving graph of startups and investors with unsupervised graph self attention, refines embeddings via link prediction and node classification losses, and feeds monthly graph snapshots into an LSTM to predict success over five years. **BERT Fusion** (Maarouf et al., 2025) concatenates BERT embeddings of each startup's Crunchbase[3] self-description with structured fundamentals and trains a lightweight neural classifier to predict success. **Standard RAG** (Lewis et al., 2020) uses a frozen dense retriever to embed queries and passages into a shared semantic space, retrieves the top $k$ similar company profiles and investor summaries, and conditions a single LLM on them via RAG Token and RAG Sequence. **SSFF** (Wang et al., 2025b) unites a divide-and-conquer multi-agent analyst block, an LLM-enhanced random-forest predictor with a founder-idea-fit network, and a RAG external-knowledge module to score startup prospects. **GNN-RAG** (Mavromatis and Karypis, 2024) couples a deep KGQA GNN that ranks candidate nodes and extracts shortest-path reasoning traces with an LLM that consumes those verbalized paths, yielding graph-aware RAG for KG question answering.

## 5.3 Evaluation Metrics

In evaluating binary classification tasks, standard metrics such as precision, recall, and F1 score are commonly used. However, to better reflect the practical needs of investors selecting high-potential startups, we adopt the Precision at $K$ ($P@K$) metric. $P@K$ measures the proportion of successful companies among the top $K$ model recommendations, where candidates are ranked by the model's predicted confidence. This metric is well-established in VC prediction research (Sharchilev et al., 2018; Zhang et al., 2021; Lyu et al., 2021) for its ability to highlight top-performing investments. Further, to assess model performance over time, we compute the Average Precision at $K$ ($AP@K$) across monthly cohorts. A higher $AP@K$ indicates that the model consistently prioritizes successful companies, thereby offering greater practical value to investors seeking to optimize portfolio decisions.

## 5.4 Parameter Settings

To prevent leakage of evaluation data into the LLM's pretraining, we use OpenAI's GPT-3.5 Turbo (knowledge cutoff: September 2021) to ensure deterministic outputs during experimentation. All training is conducted on a single NVIDIA RTX 4090 GPU with 24 GB of memory. For text embedding, we use Sentence-BERT to encode all textual inputs. Additional implementation details and parameter settings are provided in the Appendix.

## 5.5 Overall Performance Comparison

All reported metrics are averaged over five independent runs of the LLM. Table 1 shows that MIRAGE-VC improves AP@5 by 16.6%, AP@10 by 16.7%, and AP@20 by 8.0%—measures that directly capture retrieval quality when only a handful of top candidates can be pursued in practice. Notably, our model's AP@K gains increase as K decreases, demonstrating that higher confidence corresponds to

[3] Crunchbase is a public platform providing comprehensive data on companies, funding rounds, investors, and market trends.

Table 1: Performance comparison with baselines. All values are percentages (the "%" sign is omitted). *AP@K* indicates the monthly-averaged Precision@k.

| **Methods** | **AP@5** | **AP@10** | **AP@20** | **Precision** | **Recall** | **F1** |
|---|---|---|---|---|---|---|
| SHGMNN | 25.41 | 24.56 | 26.22 | 20.65 | 82.37 | 32.97 |
| GST | 26.71 | 25.71 | 27.14 | 21.75 | **83.54** | 34.51 |
| BERT Fusion | 24.67 | 26.67 | 25.33 | 23.63 | 24.95 | 24.27 |
| Standard RAG | 24.43 | 24.12 | 25.23 | 23.12 | 60.34 | 33.43 |
| SSFF | 28.23 | 30.02 | 28.42 | 23.23 | 69.41 | 34.81 |
| GNN-RAG | 29.42 | 27.53 | 27.04 | 22.81 | 71.10 | 34.54 |
| **Ours** | **34.29** | **32.14** | **29.21** | **24.32** | 73.44 | **36.54** |

greater accuracy in identifying promising startups—a property that is particularly valuable in real-world decision-making. It also achieves relative gains of 5.0% in F1 and 2.9% in Precision over the strongest baselines, indicating a more balanced and accurate classification of success outcomes. Compared to GNN-based methods (GST and SHGMNN), which boost recall through broad structural coverage but suffer precision drops from noisy neighbors; SSFF, a recent LLM-driven VC predictor that combines multi-agent analysis and RAG yet often surfaces redundant evidence; and RAG-based approaches (Standard RAG and GNN-RAG), which ground predictions in text but ignore explicit multi-hop relational chains—MIRAGE-VC's information-gain path retriever filters out low-value graph paths, and its multi-view gating adaptively weighs heterogeneous evidence, yielding balanced recall, higher precision, and a stronger ability to surface top-performing investment candidates.

## 6 Ablation Study

Table 2 reports the results of our ablation studies. Removing the graph retrieval component significantly cuts Precision by 1.3% and F1 by 2.5%, highlighting the essential role of structural evidence. Both naively concatenating all 3-hop neighbors and picking paths at random degrade performance, demonstrating the superior noise-filtering capability of our path selector. Eliminating either similar company documents or the investor profiles also results in a notable performance drop, indicating their complementary value. In the fusion stage, aggregating all evidence within a single agent costs 1.4% F1, and replacing the learnable gating mechanism with fixed weights further costs 0.6% F1, highlighting the necessity of multi-agent fusion and adaptive gating. These results collectively show that instance-level, multi-perspective reasoning combined with adaptive gating is crucial for robust performance.

Table 2: Results of ablation studies.

| **Removed Sub-module** | **Precision** | **F1** |
|---|---|---|
| w/o Graph Retrieval | 23.01 | 34.06 |
| w/o Path Selector (all) | 22.72 | 33.29 |
| w/o Path Selector (random) | 23.24 | 34.76 |
| w/o Similar Company | 23.45 | 35.54 |
| w/o Investor Analysis | 23.32 | 35.43 |
| w/o Multi-agent Fusion | 22.97 | 35.13 |
| w/o Gating Network | 24.05 | 35.94 |
| FULL | 24.32 | 36.54 |

## 7 Conclusion

This paper presents the first multi-view RAG framework for VC prediction, integrating both document-based evidence and investment relationship graphs. To address the unique challenges of VC scenarios, we propose an information gain-driven, chain-based path retrieval mechanism. This method uses a frozen LLM to estimate the marginal information gain at each graph hop and trains a lightweight path selector to efficiently extract concise, high-value reasoning chains for inference. In addition, we develop a multi-agent collaborative inference strategy coupled with an adaptive gating network that dynamically fuses diverse perspectives from textual and structural evidence. This design enables interpretable and accurate predictions without fine-tuning the underlying LLMs. Extensive experiments on real-world PitchBook datasets demonstrate that our framework outperforms multiple state-of-the-art baselines.

## Limitations

While MIRAGE-VC pioneers the integration of graph-path retrieval with RAG in VC prediction, it exhibits the following limitations:

- LLM-Driven Supervision Bias. Our path selector relies on task-specific information gains $\Delta$ computed by a frozen LLM backbone, which inherits that model's calibration errors and prompt sensitivity. Consequently, $\Delta$ may fluctuate across different LLM architectures or prompt designs. Future work should cross–validate $\Delta$ with multiple backbones, report inter–LLM agreement, and audit a stratified subset with human judgment.
- Myopic Supervision Objective. The current supervision scheme is local: $\Delta$ measures only the one-hop impact of adding a single node, and our beam search optimizes these immediate gains. This greedy strategy may overlook globally optimal subgraphs or beneficial interactions among multiple nodes. To address this, we plan to explore look-ahead scoring mechanisms and sequence-level optimization techniques—such as reinforcement learning over path sequences—to capture long-term dependencies and holistic graph structures.

## Ethical Considerations

**Data provenance and consent.** We rely exclusively on publicly available and licensed (PitchBook) company and investor level records, using identifiers only for identity resolution. No human-subject data are collected, and we do not disclose raw documents and proprietary records.

**Privacy, licensing, and compliance.** All inputs come from public or licensed fields, with strict temporal ordering to prevent future-event leakage. Investor demographics (e.g., education, gender) are used solely in aggregated summaries. Users must honor the original data licenses and must not attempt re-identification.

## References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Javier Arroyo, Francesco Corea, Guillermo Jimenez-Diaz, and Juan A Recio-Garcia. 2019. Assessment of machine learning performance for decision support in venture capital investments. *Ieee Access*, 7:124233–124243.

Brad Badertscher, Nemit Shroff, and Hal D White. 2013. Externalities of public firm presence: Evidence from private firms' investment decisions. *Journal of Financial Economics*, 109(3):682–706.

Scott Barnett, Stefanus Kurniawan, Srikanth Thudumu, Zach Brannelly, and Mohamed Abdelrazek. 2024. Seven failure points when engineering a retrieval augmented generation system. In *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering-Software Engineering for AI*, pages 194–199.

Paul Belleflamme, Thomas Lambert, and Armin Schwienbacher. 2014. Crowdfunding: Tapping the right crowd. *Journal of business venturing*, 29(5):585–609.

Francisco Ramadas da Silva Ribeiro Bento. 2017. Predicting start-up success with machine learning. Master's thesis, Universidade NOVA de Lisboa (Portugal).

Shai Bernstein, Kunal Mehta, Richard R Townsend, and Ting Xu. 2022. Do startups benefit from their investors' reputation? evidence from a randomized field experiment. Technical report, National Bureau of Economic Research.

Grahame Boocock and Margaret Woods. 1997. The evaluation criteria used by venture capitalists: evidence from a uk venture fund. *International Small Business Journal*, 16(1):36–57.

Minbyul Jeong, Jiwoong Sohn, Mujeen Sung, and Jaewoo Kang. 2024. Improving medical reasoning through retrieval and self-reflection with retrieval-augmented large language models. *Bioinformatics*, 40(Supplement_1):i119–i129.

William R Kerr, Ramana Nanda, and Matthew Rhodes-Kropf. 2014. Entrepreneurship as experimentation. *Journal of Economic Perspectives*, 28(3):25–48.

Hyungjin Ko and Jaewook Lee. 2024. Can chatgpt improve investment decisions? from a portfolio management perspective. *Finance Research Letters*, 64:105433.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wentau Yih, Tim Rocktäschel, and 1 others. 2020.

Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Jasper Linders and Jakub M. Tomczak. 2025. Knowledge graph-extended retrieval augmented generation for question answering. *Preprint*, arXiv:2504.08893.

Xiao-Yang Liu, Guoxuan Wang, Hongyang Yang, and Daochen Zha. 2023. Fingpt: Democratizing internet-scale data for financial large language models. *arXiv preprint arXiv:2307.10485*.

Hongzhen Lv, Wenzhong Yang, Fuyuan Wei, Jiaren Peng, and Haokun Geng. 2024. Mdf: A dynamic fusion model for multi-modal fake news detection. *arXiv preprint arXiv:2406.19776*.

Shiwei Lyu, Shuai Ling, Kaihao Guo, Haipeng Zhang, Kunpeng Zhang, Suting Hong, Qing Ke, and Jinjie Gu. 2021. Graph neural network based vc investment success prediction. *arXiv preprint arXiv:2105.11537*.

Abdurahman Maarouf, Stefan Feuerriegel, and Nicolas Pröllochs. 2025. A fused large language model for predicting startup success. *European Journal of Operational Research*, 322(1):198–214.

Dat Mai. 2024. Stockgpt: A genai model for stock prediction and trading. *arXiv preprint arXiv:2404.05101*.

Costas Mavromatis and George Karypis. 2024. Gnn-rag: Graph neural retrieval for large language model reasoning. *arXiv preprint arXiv:2405.20139*.

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

Junlang Qian, Zixiao Zhu, Hanzhang Zhou, Zijian Feng, Zepeng Zhai, and Kezhi Mao. 2025. Beyond the next token: Towards prompt-robust zero-shot classification via efficient multi-token prediction. *arXiv preprint arXiv:2504.03159*.

J. Ross Quinlan. 1986. Induction of decision trees. *Machine learning*, 1(1):81–106.

Ahmmad O. M. Saleh, Gokhan Tur, and Yucel Saygin. 2024. SG-RAG: Multi-hop question answering with large language models through knowledge graphs. In *Proceedings of the 7th International Conference on Natural Language and Speech Processing (ICNLSP 2024)*, pages 439–448, Trento. Association for Computational Linguistics.

Boris Sharchilev, Michael Roizner, Andrey Rumyantsev, Denis Ozornin, Pavel Serdyukov, and Maarten De Rijke. 2018. Web-based startup success prediction. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 2283–2291.

Olav Sorenson and Toby E Stuart. 2001. Syndication networks and the spatial distribution of venture capital investments. *American journal of sociology*, 106(6):1546–1588.

Xinyu Wang, Jijun Chi, Zhenghan Tai, Tung Sum Thomas Kwok, Muzhi Li, Zhuhong Li, Hailin He, Yuchen Hua, Peng Lu, Suyuchen Wang, and 1 others. 2025a. Finsage: A multi-aspect rag system for financial filings question answering. *arXiv preprint arXiv:2504.14493*.

Xisen Wang, Yigit Ihlamur, and Fuat Alican. 2025b. Ssff: Investigating llm predictive capabilities for startup success through a multi-agent framework with enhanced explainability and performance. *Preprint*, arXiv:2405.19456.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Nirmalie Wiratunga, Ramitha Abeyratne, Lasal Jayawardena, Kyle Martin, Stewart Massie, Ikechukwu Nkisi-Orji, Ruvan Weerasinghe, Anne Liret, and Bruno Fleisch. 2024. Cbr-rag: case-based reasoning for retrieval augmented generation in llms for legal question answering. In *International Conference on Case-Based Reasoning*, pages 445–460. Springer.

Junchi Yu, Tingyang Xu, Yu Rong, Yatao Bian, Junzhou Huang, and Ran He. 2021. Recognizing predictive substructures with subgraph information bottleneck. *IEEE transactions on pattern analysis and machine intelligence*, 46(3):1650–1663.

Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yuechen Jiang, Yupeng Cao, Zhi Chen, Jordan Suchow, Zhenyu Cui, Rong Liu, and 1 others. 2024. Fincon: A synthesized llm multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. *Advances in Neural Information Processing Systems*, 37:137010–137045.

Liangliang Zhang, Zhuorui Jiang, Hongliang Chi, Haoyang Chen, Mohammed Elkoumy, Fali Wang, Qiong Wu, Zhengyi Zhou, Shirui Pan, Suhang Wang, and 1 others. 2025. Diagnosing and addressing pitfalls in kg-rag datasets: Toward more reliable benchmarking. *arXiv preprint arXiv:2505.23495*.

Shengming Zhang, Hao Zhong, Zixuan Yuan, and Hui Xiong. 2021. Scalable heterogeneous graph neural networks for predicting high-potential early-stage startups. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2202–2211.

Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, and 1 others. 2024. A multimodal foundation agent for financial trading: Tool-augmented, diversified, and generalist. In *Proceedings of the 30th acm sigkdd conference on knowledge discovery and data mining*, pages 4314–4325.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Han Zhou, Xingchen Wan, Lev Proleev, Diana Mincu, Jilin Chen, Katherine Heller, and Subhrajit Roy. 2023. Batch calibration: Rethinking calibration for in-context learning and prompt engineering. *arXiv preprint arXiv:2309.17249*.

# A Appendix

## A.1 More Related Work

With the rise of LLMs, a variety of LLM-based financial and VC decision-support systems have been proposed (Liu et al., 2023; Zhang et al., 2024; Ko and Lee, 2024). These systems typically rely on textual and numerical features or on simulation of real-world scenarios to improve prediction accuracy. For example, SSFF (Wang et al., 2025b) and FinCon (Yu et al., 2024) systems establish a hierarchical manager–analyst collaboration mechanism, outperforming expert teams across multiple tasks; and StockGPT (Mai, 2024) is pretrained on extensive quantitative stock-market data to autonomously learn price-movement patterns, yielding substantial excess returns and demonstrating the promise of generative AI in complex financial decision making. However, none of these approaches directly integrate graph-structured knowledge—such as investor–startup relationship networks—and thus they are unable to fully capture path dependencies.

Some KG-RAG approaches (Saleh et al., 2024; Linders and Tomczak, 2025) retrieve relevant triples or subgraphs, linearize them into text, and condition an LLM for factoid question answering. These frameworks excel at matching and verifying discrete facts but lack any mechanism to filter or rank multi-hop paths by their downstream utility. In contrast, our VC prediction task requires selecting concise, high-value investment chains as explicit evidence for node-classification inference, so standard KG-RAG methods cannot be directly applied without substantial adaptation.

## A.2 Impact of Parameters on Performance

We randomly sampled 1,000 companies from the non-test portion of our PitchBook data and, for each hyperparameter setting, ran each perspective's retrieval-and-analysis component five times to estimate mean $F_1$ scores and standard errors (Figure 4). In the document retrieval study (Figure 4 (a)), we varied the number of similar-company shots $K$ and lead-investor résumé entries $N$ from 0 to 6: performance rose sharply above the zero-shot baseline, peaked at $(K, N) = (4, 5)$, and then plateaued, motivating our choice of $K = 4$, $N = 5$. In the graph retrieval grid (Figure 4 (b)), we swept maximum search depths $d_{\max} \in \{1, \dots, 6\}$ and path counts $P \in \{1, \dots, 4\}$; the highest $F_1$ was observed at $(P, d_{\max}) = (2, 4)$, which we adopt in all main experiments.

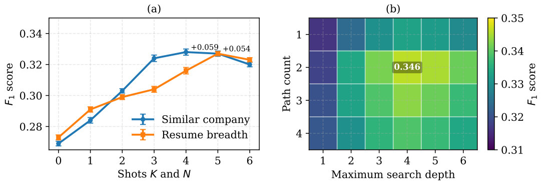

Figure 4: (a) Document retrieval: effect of the number of similar companies $K$ and resume breadth $N$ on the agents' $\mathrm{F}_1$. (b) Graph retrieval: effect of search depth $d_{\max}$ and path count $P$ on $\mathrm{F}_1$.

## A.3 Implementation Details

### A.3.1 Path Selector

**Binary probability of LLMs** We follow the mainstream likelihood-based scoring practice that normalizes the token-level log-likelihoods of verbalized labels (e.g., True/False) to obtain a Bernoulli probability, as adopted and analyzed in recent work on zero-shot classification, calibration, and probability-based prompt selection (Zhou et al., 2023; Qian et al., 2025). Given a prompt $P$, we verbalize labels as the strings "True" (Success) and "False" (Failure). For a string $w = (t_1, \dots, t_m)$, we use the string log-likelihood

$$\log P(w \mid P) = \sum_{j=1}^{m} \log P(t_j \mid P, t_{<j}) \,.$$

Let

$$L_T = \log P(\text{“True”} \mid P), L_F = \log P(\text{“False”} \mid P)$$

The success probability is obtained by two-way normalization:

$$p = \frac{e^{L_T}}{e^{L_T} + e^{L_F}} = \sigma\big(L_T - L_F\big), \quad \sigma(x) = \tfrac{1}{1+e^{-x}}$$

We only query log-probabilities for the target strings.

**Training Settings** The trade-off weight in Eq. (2), $\lambda_{\text{conf}} \in [0, 1]$, is selected on the validation split by a small grid search; Table 3 shows that performance is stable in the range $0.1 - 0.3$, and we therefore fix $\lambda_{\text{conf}} = 0.2$ in all experiments. Hyperparameter is listed in Table 6.

Table 3: Validation *NDCG@1* (%) under different confidence–trade-off weights $\lambda_{\text{conf}}$.

| $\lambda_{\text{conf}}$ | 0 | 0.10 | 0.20 | 0.30 | 0.40 | 0.50 |
|---|---|---|---|---|---|---|
| NDCG@1 (%) | 62.7 | 63.1 | **63.4** | 63.2 | 62.9 | 62.5 |

**Evaluation and Results** We evaluate the Path Selector with two ranking metrics. Hit@1 measures whether the selector's top-ranked candidate matches an oracle best-gain candidate in the group. It directly reflects the success rate of our top-1 expansion policy at inference, thus aligning tightly with how the selector is actually used; a higher Hit@1 means we more often choose the maximal-gain extension. NDCG@1 normalizes the gain of the selected candidate by the maximum attainable gain in the group, yielding a score in $[0, 1]$. Unlike the binary Hit@1, NDCG@1 gives partial credit when the chosen candidate is near-optimal, making it more stable under noisy or close-valued oracle gains and better for hyperparameter tuning. A random baseline is obtained by drawing an i.i.d. score from $\mathcal{U}(0, 1)$ for every candidate and applying the same evaluation procedure. Compared with random scoring, the Path Selector improves Hit@1 by $+0.0879$ and NDCG@1 by $+0.1851$, confirming its ability to consistently priorities expansions of higher task-specific information gain.

Table 4: Selector performance on the held-out TEST split.

| Method | NDCG@1 (%) | Hit@1 (%) |
|---|---|---|
| Random $\mathcal{U}(0, 1)$ | 44.92 | 33.33 |
| PATH SELECTOR | **63.43** | **42.12** |

## A.4 Gate Network

**Data Preparation** Each training instance consists of the three agent rationales, encoded by a frozen all-MiniLM-L6-v2 sentence encoder into 384-dimensional vectors $\{\mathbf{h}_i\}_{i=1}^{3}$. Static company descriptors (industry, region, funding round) are one-hot-encoded into a 14-dimensional vector $\mathbf{c}$ and concatenated with the agent embeddings.

**Training Settings** Query–key projections $\mathbf{W}_Q, \mathbf{W}_K$ are applied to each $\mathbf{h}_i$ to obtain view-level attention scores. The attended view representations, together with $\mathbf{c}$, are fed to a two-layer MLP that outputs instance-specific weights $\mathbf{w} \in \mathbb{R}^3$ ($\sum_i w_i = 1$). The weighted sum of the three views is finally mapped by a second two-layer MLP to the success probability. All remaining hyper-parameters are summarized in Table 6.

Table 5: Performance of the gating network versus a random-weight baseline on the held-out test set.

| Method | P (%) | F1 (%) |
|---|---|---|
| Random | 19.98 | 19.49 |
| Ours | **23.15** | **35.13** |

**Evaluation and Results** Following the main task, the gating network is assessed on Precision (P) and F1. As a sanity check we replace the learned weights by a uniform random choice (Random); its performance marks the chance level of selecting the most informative view. Table 5 shows that the learned gate substantially outperforms this baseline, confirming that the network has indeed captured non-trivial view–attribute interactions.

Table 6: Hyper-parameters for the Gate Network (GN) and Path Selector (PS)

| Parameter | GN | PS |
|---|---|---|
| Text vector dimension | 384 | 384 |
| Company key dimension | 53 | — |
| Batch size | 256 | 256 |
| Training epochs | 50 | 30 |
| Hidden width | 256 | 256 |
| Optimiser | AdamW | AdamW |
| Learning rate | $5 \times 10^{-4}$ | $3 \times 10^{-4}$ |
| Temperature $\tau$ | — | 0.5 |

## A.5 Text Embedding Model Analysis

Two pipeline components rely on a frozen sentence-encoder to obtain text representations: (i) the *graph retriever*, where the encoder embeds path descriptions, and (ii) the *gate network*, where it embeds each agent's generated answer. Here we analyse whether swapping the encoder markedly affects intermediate metrics. Tables 7 and 8 show that across both modules the performance gap between alternative encoders is marginal, confirming that our main results are not sensitive to the specific choice of text-embedding model.

Table 7: Impact of different text encoders on the **graph retriever**.

| Text encoder | Dim. | NDCG@1 (%) | Hit@1 (%) |
|---|---|---|---|
| all-MiniLM-L6-v2 | 384 | 63.3 | 42.2 |
| jina-embeddings-v2-base | 768 | 63.4 | 42.1 |
| e5-large-v2 | 1024 | 63.1 | 41.9 |

Table 8: Impact of different text encoders on the **gate network**.

| Text encoder | Dim. | $P$ (%) | $F_1$ (%) |
|---|---|---|---|
| all-MiniLM-L6-v2 | 384 | 23.45 | 35.13 |
| jina-embeddings-v2-base | 768 | 23.12 | 35.31 |
| e5-large-v2 | 1024 | 23.25 | 35.25 |

### A.6 Resource Utilization and Latency

Our end-to-end pipeline comprises three phases: (i) retrieval via GPT-3.5 API, (ii) oracle scoring with Llama-3.1-8B and selector training, and (iii) gating network training and ablation studies. We issued approximately 40 000 GPT-3.5 requests (under 12 000 tokens each), processing ≈ 480 million tokens (under 48 GPU hour). Locally, we generated 16 857 gain labels with Llama-3.1-8B (5 619 group expansions, all within its 8 000-token window). Training the listwise selector on an NVIDIA RTX 4090 (24 GB VRAM) for 50 epochs required about 5 minutes, and the gating network plus ablations completed in under 10 minutes. Including hyperparameter sweeps, total GPU time was 10 GPU h. At inference, producing three retrieval views via GPT-3.5 followed by the adaptive fusion pass completes in under 3 minutes per company, demonstrating the pipeline's efficiency and suitability for low-latency decision support.

### A.7 Prompt Templates and Examples

This section provides the exact prompt templates used by each module and one illustrative example per template.

#### A.7.1 Company and Investor Basic Info Case

```
### Company Profile ###
Company name    : ACME Robotics
Founded year    : 2023
Headquarters    : San Francisco, USA
Industry        : Service Robotics
Employees       : 35 (as of 2025)
Key prototype   : Compact autonomous cleaning
    robot for boutique hotels
Revenue status  : Pre-revenue; paid pilots
    scheduled Q4-2025
Funding to date : USD 3.5 M (Seed round, Jun
    -2024)
Lead investors  : FutureFund (Jane Doe),
    SeedSpark Ventures
Company overview: ACME Robotics develops AI-
    driven service robots that automate
    routine cleaning tasks in hospitality and
    small retail environments. The platform
    combines low-cost modular hardware with on
    -device perception and a subscription
    software stack, aiming to deliver pay-as-
    you-go automation for venues that cannot
    afford traditional industrial solutions.
```

```
### Lead-Investor Profile ###
Investor name: Jane Doe  Partner @ FutureFund
Tenure       : 2016 - present

Previous positions
• Senior Engineer, ABB Robotics (2008 - 2012)
    global industrial-robotics leader.{
    COMPANY_PROFILE} (success)
• Investment Associate, TechEdge Capital (2012
     - 2016)early-stage deep-tech VC.\{COMPANY
    \_PROFILE\} (success)

Focus sectors    : Robotics • Edge AI • IoT
Assets under mgmt: USD 1.4 B

Investment record
• RoboVacacquired by Dyson (2021). {
    COMPANY_PROFILE} (success)
• MechArmIPO (2022). {COMPANY_PROFILE} (
    success)
• NanoGripacquired by Bosch (2020). {
    COMPANY_PROFILE} (success)
• ServoLinkceased operations (2019). {
    COMPANY_PROFILE} (failure)

Board seats : MechArm • FlexDroid • SensorX
Awards      : Forbes 30 Under 40 in VC (2023)
```

#### A.7.2 Path Analyst Prompt

```
Role: You are a senior venture-capital analyst
     who excels at step-by-step
      reasoning over investment paths to judge
     whether a seed / angel-stage
      start-up is likely to secure Series-A
    funding within the next year.

You are given three blocks of information:

(1) High-value investment path retrieved for {
    COMPANY_NAME}:{PATH_TEXT}
(2) Company profiles appearing in the path (
    each with outcome labels; True = raised
    Series A
    within 12 months after seed/angel, False =
     did not):{COMPANY_PROFILES} Success/
    Failure:{LABELS}
(3) Investor profiles appearing in the path:{
    INVESTOR_PROFILES}
(4) Target company profile:{
    TARGET_COMPANY_PROFILE}
Task:
  • Analyse the evidence and predict whether {
    COMPANY_NAME} will
    raise a Series-A round within 12 months.
  • Output **exactly** in the format:

      Prediction: True/False
      Analysis: <your step-by-step reasoning>

  • If evidence is insufficient, reason
    cautiously but still decide.
```

### A.7.3 Company Analyst Prompt

```
Role:
  You are a senior venture-capital analyst who
      excels at using information from
  industry peers (similar companies) to judge
    whether a seed/angel-stage target
  will secure Series-A funding within the next
      year.

You are given:

(1) Target company profile:{
    TARGET_COMPANY_PROFILE}

(2) Comparable companies (each with outcome
    labels; True = raised Series A
    within 12 months after seed/angel, False =
     did not):{COMPANY_PROFILES} Success/
    Failure:{LABELS}

Task:
  • Analyse the evidence and predict whether {
    COMPANY_NAME} will
    raise a Series-A round within 12 months.
  • Output **exactly** in the format:

      Prediction: True/False
      Analysis: <your step-by-step reasoning>

  • If evidence is insufficient, reason
    cautiously but still decide.
```

### A.7.4 Investor Analyst Prompt

```
Role:
  You are a senior venture-capital analyst who
      specialises in evaluating a
  start-ups lead seed/angel investor record to
      judge whether the target can
  secure Series-A funding within the next year.


You are given:
(1) Target company profile:{
    TARGET_COMPANY_PROFILE}

(2) Lead-investor rsum (prior operating roles
    and portfolio companies,
    each annotated as success or
    failuresuccess = the company raised Series
     A
    within 12 months of its seed/angel round;
    failure = it did not):{INVESTOR_PROFILE}

Task:
  • Analyse how the investors past successes
    and failures relate to the target
    companys sector, stage, and needs.
  • Predict whether the target will raise a
    Series-A round within 12 months.
  • Output **exactly** in the format:

      Prediction: True/False
      Analysis: <your step-by-step reasoning>

  • If evidence is insufficient, reason
    cautiously but still decide.
```

### A.7.5 Manager Analyst Prompt

```
Role:
  You are a senior venture-capital analyst who
      excels at synthesizing other
  experts viewpoints to decide whether a seed/
    angel-stage start-up will secure
  Series-A funding within the next year.

You are given:

(1) Path-analyst verdict
    • Prediction: {PATH_PREDICTION}
    • Analysis  : {PATH_ANALYSIS}

(2) Similar-company analyst verdict
    • Prediction: {SIM_PREDICTION}
    • Analysis  : {SIM_ANALYSIS}

(3) Lead-investor analyst verdict
    • Prediction: {INV_PREDICTION}
    • Analysis  : {INV_ANALYSIS}

(4) Aggregate-weight advice
    The historical importance of the three
    perspectives is
    {WEIGHTS_VECTOR}

(5) Target company profile
    {TARGET_COMPANY_PROFILE}

Task:
  • Produce a single, final prediction on
    whether the target will raise a
    Series-A round within 12 months.
  • Output **exactly** in the format:

      Prediction: True/False
      Analysis: <your step-by-step reasoning>

  • If evidence is insufficient, reason
    cautiously but still decide.
```