Analyze Like a Venture Capitalist: Information-Gain and Knowledge Enhanced Graph Reasoning for Startup Success Prediction

Anonymous ACL submission

Abstract

Most venture capital (VC) investments fail, while a few deliver outsized returns. Predicting startup success requires synthesizing relational evidence across company fundamentals, investor track records, and investment networks through explicit reasoning, which traditional machine learning and graph neural networks lack. Large language models excel at reasoning, but applying them to VC prediction must address: selecting compact evidence subgraphs from large investment networks, one-sided label noise where failures may be latent successes, and grounding decisions in structured VC domain knowledge. We present MIRAGE-VC, an evidence-grounded reasoning framework with three innovations. First, an information-gain-driven retriever distills networks into compact evidence subgraphs. Second, a dual-layer knowledge base grounds reasoning in VC principles. Third, a noise-aware mechanism down-weights mislabeled negatives via improved Positive-Unlabeled (PU) estimation. MIRAGE-VC achieves +5.9% F1 and +22.1% Precision@5 over state-of-the-art baselines. Expert evaluation confirms professional-quality rationales. We further validate our approach on public data with consistent improvements. Code and reasoning results available.[1]

1 Introduction

Venture capital (VC) investment is characterized by extreme asymmetry: 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). This high-risk, high-reward profile makes accurate startup success prediction crucial for portfolio optimization and capital allocation.

The challenge is to synthesize relational evidence across sources (Gompers et al., 2020). A typical decision integrates (i) peer-relative market positioning (Kim and Ritter, 1999), (ii) lead investors' track record and reputation (Hsu, 2004), and (iii) structural signals in the investment network (co-investment patterns, coalitions) (Hochberg et al., 2007). For WhatsApp, the subgraph "WhatsApp ← Sequoia → Google ← Kleiner Perkins → Amazon" (Figure 1) suggests top-tier screening, adjacency to tech giants, and strong VC coalitions that can ease follow-on financing and strategic M&A. Turning such heterogeneous signals into coherent, interpretable theses requires *explicit reasoning* (Kaplan and Strömberg, 2004).

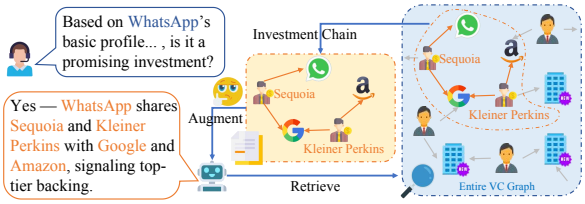

Figure 1: Impact of selected subgraph on prediction.

Existing approaches fall short. Traditional methods (Arroyo et al., 2019; Bento, 2017) use isolated firm features and ignore relational context. GNNs (Lyu et al., 2025; Zhang et al., 2021) capture higher-order investor–company structure but remain opaque without exposing reasoning, limiting interpretability and making it hard to incorporate external knowledge beyond the training graph.

Large language models (LLMs) offer a natural remedy through their strong reasoning and broad world knowledge (Liu et al., 2023; Ko and Lee, 2024). However, LLMs face a fundamental *modality mismatch* when processing graph-structured data (Wang et al., 2023): their architectures are optimized for sequential text, not relational topologies. While recent work (Sun et al., 2023; Luo et al., 2023) integrates LLMs with knowledge graphs for *in-graph question answering*, where answers lie within the graph, VC prediction is an *off-graph* task: the prediction target exists outside

[1] https://anonymous.4open.science/r/MIRAGE-VC-323F

the network, and the graph serves solely as retrievable evidence. This distinction applies to other domains such as recommendation systems (predicting user–item affinity from interaction graphs) and credit risk assessment (predicting default from transaction networks). The core challenge thus becomes: ***how to select select evidence subgraphs that maximize a predictor's performance on an external objective***, rather than finding chains that terminate at an in-graph answer.

Even restricting to path-based retrieval, the primary obstacle is **exponential path explosion**: shallow 1-hop neighborhoods around a target company often lack sufficient signal (Yu et al., 2021), yet extending to 3 or 4 hops generates thousands of candidate paths, many redundant or weakly informative (Zhang et al., 2025). Indiscriminately feeding all paths to an LLM overwhelms its context window and dilutes attention.

A second challenge stems from **incomplete outcome tracking in private markets**. Unlike public equities, venture exits may go unrecorded through delayed IPOs, quiet acquisitions, or sustained profitability (Giot and Schwienbacher, 2007; Kaplan and Lerner, 2016). This creates ***one-sided label noise***: observed successes are reliable, but observed "failures" often mean "no exit yet" rather than true failure, conflating genuine negatives with *latent successes*. Standard supervised learning systematically penalizes companies with delayed exits (Frénay and Verleysen, 2013; Song et al., 2022).

Beyond these technical obstacles, real-world VC decisions require **domain expertise** rarely encoded in local datasets. While our predictor observes the target company's profile and relational neighborhood, it lacks broader knowledge such as valuation heuristics, stage-specific risk patterns, and empirically grounded investment theses (MacMillan et al., 1985; Gompers, 2022), risking overfitting to idiosyncratic graph patterns.

We propose MIRAGE-VC, an evidence-grounded reasoning framework with:

Information-gain-driven path retrieval. To tackle path explosion, we introduce a retriever that iteratively expands from the target company, at each step selecting the neighbor that maximally improves an LLM predictor's accuracy. By aggregating these high-value neighbors across multiple steps, we extract compact subgraphs composed of overlapping investment chains rather than isolated paths. This selector is trained offline using task-specific information gain signals.

Dual-layer VC domain knowledge base. To bridge the domain expertise gap, we construct an external retrieval corpus with (i) a *theory layer* distilled from authoritative books and academic studies, and (ii) a *practice layer* emphasizing actionable playbooks. Retrieved passages are provided as structured guidance alongside local evidence.

Noise-aware training with PU-based estimation. To address one-sided label noise, we train a compact manager via supervised fine-tuning followed by *noise-aware Direct Preference Optimization (DPO)* (Rafailov et al., 2023). DPO trains the model to prefer investment rationales that led to successful exits over those associated with failures. However, since some observed "failures" are actually latent successes, we use Positive-Unlabeled (PU) learning (Kiryo et al., 2017) to estimate each negative sample's probability of being a latent success, and down-weight its penalty in DPO updates.

We explicitly assess the *interpretability* of our generated rationales using a dual evaluation protocol: human VC-expert review complemented by an LLM-based rubric. This strengthens the claim that our method improves not only predictive accuracy but also transparency and auditability.

Our contributions are as follows:

- We propose an **information-gain-driven** method that distills VC networks into compact, high-value subgraphs and integrates **VC knowledge** (theory and practice), enabling explicit, human-auditable reasoning over relational patterns and investment principles.
- We introduce a **noise-aware training strategy** using PU-based unreliability estimation and DPO to address one-sided label noise in private-market outcomes.
- We achieve **state-of-the-art** on real-world VC data (+5.9% F1, +22.1% Precision@5), with consistent improvements validated on public data. VC expert evaluation confirms our rationales achieve professional-quality investment reasoning. Our paradigm of selecting graph evidence by marginal utility generalizes to other off-graph prediction tasks. We make the code and reasoning results available.

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., 2025) 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 Graph-augmented LLMs

Recent systems couple LLMs with knowledge graphs for *in-graph* QA—answers are entities in graph, reached or verified via triple retrieval and path reasoning (Sun et al., 2023; Luo et al., 2023). While effective at fact verification, they optimize in-graph objectives (entity/relation correctness) rather than selecting multi-hop evidence by its marginal utility to an external predictor. In our off-graph VC setting, the graph serves as evidence for startup-success prediction; what is needed is utility-aware selection of a few high-value investment chains as explicit evidence for prediction.

Proposed to remedy text-only RAG's inability to model structure and multi-hop dependencies, GNN-RAG frameworks retrieve relevant nodes via embedding similarity and inject local structural cues before handing context to an LLM (Mavromatis and Karypis, 2024). Yet they typically surface no explicit reasoning paths, limiting LLMs' strength in stepwise, interpretable chain-of-thought reasoning (Wei et al., 2022) and constraining multi-hop inference over heterogeneous investment networks.

3 Preliminary

3.1 Problem Definition

This study aims to predict the success of early-stage startups, 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, 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 a startup will secure subsequent financing within one year of its initial funding.

3.2 Data Overview

PitchBook. We use the PitchBook[2] Global VC dataset as the main data, 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. See Appendix A.7 for details.

Crunchbase. To facilitate reproducibility, we use the publicly available Crunchbase 2013 snapshot[3] containing 196,553 companies and 226,708 individuals funded between 2005–2013. Despite smaller, it preserves the same relational structures.

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 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, MIRAGE-VC centers on a *manager* model that takes multi-source evidence

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

[3]https://github.com/mtwilliams/crunchbase-in-2013

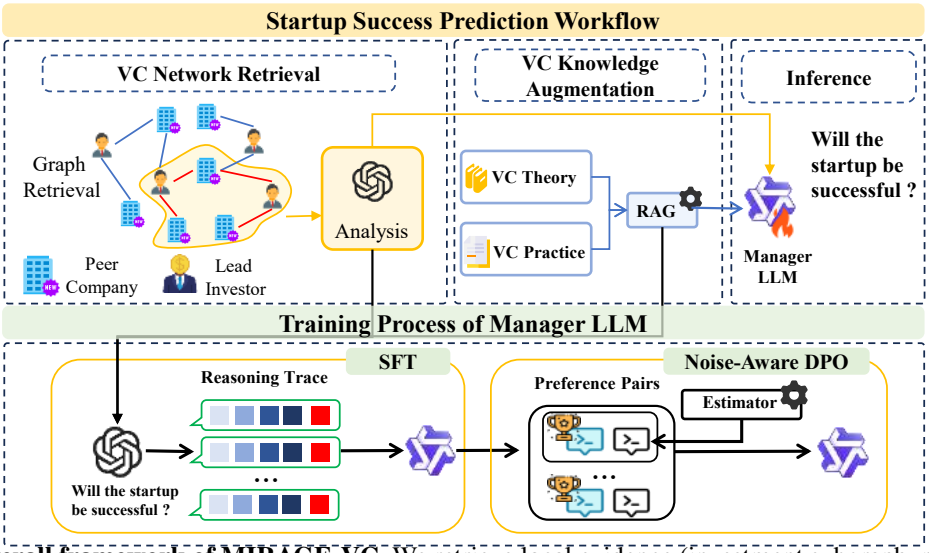

Figure 2: **Overall framework of MIRAGE-VC.** We retrieve local evidence (investment subgraph, peer company, lead investor), augment it with dual-layer VC knowledge via RAG, and train a manager with SFT followed by noise-aware DPO to produce the final prediction.

as input and outputs a binary investment decision with a structured rationale memo. For each target company, we fuse three internal evidence channels: (i) a budgeted evidence subgraph from the VC investment network, (ii) peer-company context from textual corpora, and (iii) a lead-investor profile (demographics, career history, deal records). Crucially, we enhance this evidence with external VC knowledge retrieved from a two-layer VC knowledge base (investment theories + best practices) to guide the manager's reasoning. The manager is trained with SFT for schema-following and evidence grounding, then noise-aware preference optimization that down-weights potentially mislabeled negatives via a PU-based unreliability estimator.

4.2 Graph Retrieval

4.2.1 From Classic IG to Subgraph Expansion

As shown in Figure 3, we formulate evidence retrieval as sequential *evidence subgraph expansion*. Starting from the target node c^*, at each step we select one node from the current frontier and add it to the selected set, inducing an incrementally growing evidence subgraph. We score each candidate by its task-specific marginal utility, analogous to information gain (Quinlan, 1986).

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

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.

4.2.2 LLM-generated Gain Labels

To obtain oracle supervision for the subgraph selector, we use a frozen LLM to quantify task-specific information gain per 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. During labeling, a retrieval state is represented by two sets: the selected node set $V^{(t)}$ (initially $V^{(0)} = \{c^*\}$) and the frontier set $F^{(t)}$ consisting of all 1-edge reachable nodes from $V^{(t)}$ that are not yet selected. At step t, we sample up to three candidates $\{v_1, v_2, v_3\} \subseteq F^{(t)}$ and additionally include a special STOP option $v_0 = \text{STOP}$.

Prompt Construction To measure the incremental value of each candidate node v_i, we generate two prompts per expansion: (i) a *baseline* prompt P_{base} that verbalizes the current selected subgraph induced by $V^{(t)}$, and (ii) a *candidate* prompt P_{v_i} that verbalizes the updated subgraph induced by $V^{(t)} \cup \{v_i\}$. For the STOP option, we set $P_{v_0} \equiv P_{\text{base}}$. A frozen LLAMA-3.1-8B classifier returns the success probabilities p_{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 of including a candidate node v_i as the reduction in task loss after augmenting the evidence with v_i:

$$\Delta_{v_i} = \mathrm{CE}(y,\, p_{\text{base}}) - \mathrm{CE}(y,\, p_{v_i}) \tag{2}$$

where p_{base} is the predicted success probability using the current evidence, and p_{v_i} is the prediction after adding evidence associated with v_i. CE denotes binary cross-entropy; thus $\Delta_{v_i} > 0$ indicates that including v_i improves prediction correctness under the end task.

We assign the STOP option a fixed gain $\Delta_{\text{STOP}} = 0$, which serves as a natural threshold: the retriever continues expanding only when at least one candidate yields a positive marginal gain, and terminates otherwise.

Training Tuples Each training instance is a tuple $\big(G^{(t)}, G^{(t)}_{v_i}, \Delta_{v_i}\big)$, where $G^{(t)}$ is the current evidence subgraph and $G^{(t)}_{v_i}$ is the updated subgraph after adding v_i (unchanged if $v_i = \text{STOP}$). The selector learns to predict Δ_{v_i} from $\big(G^{(t)}, G^{(t)}_{v_i}\big)$.

Since Δ_{v_i} is computed using gold labels for both successes and failures, the selector is trained to prefer expansions that reduce end-task loss under a fixed budget.

4.2.3 Selector Training Objective

Each step t of a target company contributes one *ranking group* $G^{(t)}_{\text{cand}} = \{v_0{=}\text{STOP}, v_1, v_2, v_3\}$ with associated gains $\Delta_{v_0}, \Delta_{v_1}, \Delta_{v_2}, \Delta_{v_3}$ annotated as in Eq. (2). For each candidate $v \in G^{(t)}_{\text{cand}}$, we compute a difference feature:

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

where e_{base} and e_v are 768-dimensional sentence embeddings extracted once by a frozen encoder. For STOP, we set $e_{v_0} = e_{\text{base}}$, making its feature a zero-change reference. A lightweight 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

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

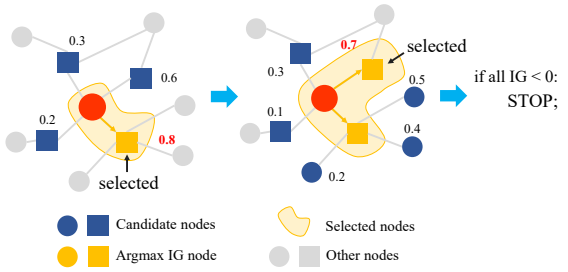

Figure 3: Illustration of how the subgraph selector retrieves the evidence from graph

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:

$$\mathcal{L}(\theta) = \sum_t \mathcal{L}_{\text{list}}(G^{(t)}_{\text{cand}}) \quad (7)$$

As shown in Figure 3, at inference, the selector greedily expands a budgeted evidence subgraph by selecting the top-scoring frontier node or STOP. See Appendix A.8.1 for more details.

4.2.4 Auxiliary Evidence Construction

To complement graph-derived evidence, we construct two auxiliary textual views—*peer-company* and *lead-investor*—and append their concise analyses to the manager input. The peer-company view retrieves top-k semantically similar firms (with temporal leakage control) and summarizes shared patterns. The lead-investor view identifies the lead investor in the first round and summarizes pre-t_0 biography and deal history. Both views are rendered in the same schema and are used as additional evidence for the manager.

4.3 External VC Knowledge Augmentation

Local evidence in the VC dataset often lacks the ***VC domain knowledge*** needed to interpret signals. We construct a two-layer VC knowledge base to ground the manager's reasoning.

VC prior knowledge can be broadly divided into (i) stable, widely applicable principles and (ii) action-oriented diligence heuristics (Gompers et al., 2020). We build two corresponding layers, curating both by topical relevance and citation frequency with ***validation by a senior VC expert***:

Layer 1: VC Theory — 15 seminal texts (avg. 503 pages, 260 citations) on investment mechanisms, e.g., Gompers (2022) and Zider (1998).

Layer 2: VC Practice — 40 practitioner playbooks (avg. 74 pages, 560 citations) on diligence frameworks, e.g., MacMillan et al. (1985) and Gompers et al. (2020).

Following RAPTOR (Sarthi et al., 2024), we employ coarse-to-fine retrieval: documents are chunked, embedded, and clustered with LLM-generated summaries. At inference, we retrieve relevant cluster summaries based on the query (company profile, investor profile, network evidence), then fetch top-n chunks and append them to the manager prompt. Details in Appendix A.3.

4.4 Manager: Reasoning-Enhanced Predictor

We train a manager to directly output a task-aligned decision that includes (i) a binary outcome prediction and (ii) an evidence-grounded, structured rationale. To this end, we adopt a two-stage optimization recipe: supervised fine-tuning (SFT) for schema-following, followed by a noise-aware preference optimization that explicitly accounts for one-sided label noise via an improved PU-based unreliability estimator and a weighted DPO objective.

Training Data Construction. For each company, we form a manager input prompt x by concatenating: (i) the company profile, (ii) retrieved internal evidence (e.g., the selected investment subgraph), (iii) auxiliary analyses from complementary views (e.g., peer-company and lead-investor analyses), and (iv) optionally retrieved external VC knowledge. We sample the base manager K times to obtain candidate outputs $\{z_k\}_{k=1}^{K}$, and construct preference pairs (x, z^+, z^-) where z^+ is preferred over z^- under our automatic rubric (format compliance, explicit evidence referencing, and internal consistency).

4.4.1 Stage 1: Supervised Fine-Tuning (SFT).

We first perform SFT on the manager using only the preferred responses z^+, minimizing the standard negative log-likelihood:

$$\mathcal{L}_{\text{SFT}} = -\,\mathbb{E}_{(x,z^+)}\big[\log \pi_\theta(z^+ \mid x)\big] \tag{8}$$

SFT teaches the manager to follow the output schema and to ground explanations in the provided evidence.

4.4.2 Estimating Instance Unreliability.

As discussed earlier, private-market outcomes are often incomplete, leading to one-sided noise: observed positives are relatively reliable, whereas observed negatives may contain latent successes. As a result, preference pairs from observed-negative instances can be unreliable, and applying DPO uniformly may overfit to such noise.

We therefore treat observed positives ($y_{\text{obs}} = 1$) as reliable positives and observed negatives ($y_{\text{obs}} = 0$) as *unlabeled*. For each observed-negative instance, we estimate an unreliability score $q(x) \in [0, 1]$:

$$q(x) \approx \Pr(y = 1 \mid x,\ y_{\text{obs}} = 0) \tag{9}$$

which reflects the likelihood that an observed negative is a latent positive given its evidence context.

Following PU learning, we first train a probabilistic classifier

$$g(x) \approx \Pr(s = 1 \mid x) \tag{10}$$

where $s = 1$ denotes an *observed* positive and $s = 0$ denotes unlabeled. Classical PU assumes a constant propensity $c = \Pr(s = 1 \mid y = 1)$ (SCAR), giving $\Pr(y = 1 \mid x) \approx g(x)/c$. In our setting, however, the chance that a truly successful company is *observed* as success depends on instance attributes and coverage, so the propensity is better modeled as an instance-dependent function

$$c(x) = \Pr(s = 1 \mid y = 1, x) \tag{11}$$

which captures how likely a true positive is recorded as an observed positive for a specific instance.

We thus estimate an effective $\hat{c}(x)$ and define

$$q(x) = \mathbb{I}[y_{\text{obs}}{=}0] \cdot \min\left(1,\ \frac{g(x)}{\hat{c}(x)}\right) \tag{12}$$

so that $q(x)$ is *only* used to quantify unreliability among observed negatives.

Estimating $\hat{c}(x)$. We calibrate $\hat{c}(x)$ using *out-of-fold* predictions and a *high-confidence* calibration split. Concretely, within each stratum/bucket b, we choose a bucket-level propensity $\hat{c}_b$ so that the *implied* latent-success rate among observed negatives in the high-confidence region matches the observable anchor $1 - \widehat{\text{P@K}}_b$ (solved by a 1D search; Appendix A.8.2). We then implement $\hat{c}(x)$ as a piecewise function $\hat{c}(x) = \hat{c}_{b(x)}$ over coarse instance strata.

4.4.3 Stage 2: Noise-Aware DPO.

We incorporate $q(x)$ into preference optimization by down-weighting updates from potentially mislabeled negatives. Crucially, $q(x)$ affects *only* observed negatives ($y_{\text{obs}} = 0$): it reduces the gradient magnitude of their preference loss, mitigating the influence of latent-success negatives on preference learning.

Let $w(x) \in (0, 1]$ be an instance weight:

$$w(x) = \max\Big(w_{\min},\ 1 - \mathbb{I}[y_{\text{obs}}{=}0] \cdot q(x)\Big) \quad (13)$$

where $w_{\min}$ prevents vanishing gradients. Given a preference pair (x, z^+, z^-) and a reference policy π_{ref} (initialized from the SFT policy), we optimize the weighted DPO objective:

$$\Delta_\theta(x) = \log \frac{\pi_\theta(z^+ \mid x)}{\pi_\theta(z^- \mid x)} - \log \frac{\pi_{\text{ref}}(z^+ \mid x)}{\pi_{\text{ref}}(z^- \mid x)} \quad (14)$$

$$\mathcal{L}_{\text{NA-DPO}} = -\,\mathbb{E}\Big[w(x)\ \log \sigma\big(\beta\, \Delta_\theta(x)\big)\Big] \quad (15)$$

where β controls the preference optimization.

5 Experiments & Results

5.1 Datasets

1. Selector training split. We train the subgraph selector on a sampled subset of 2,000 companies from the full VC investment graph, preserving the overall success-to-failure ratio. We split this subset into train/val/test by 70:15:15 with class balance.
2. Manager training/validation and final test. We adopt a temporally ordered split with an overall train/val/test ratio of 7:1:2 for both PitchBook and Crunchbase, such that training/validation instances precede the test period. For PitchBook, the dataset contains 9,410 instances in total. For Crunchbase, the dataset contains 4,375 instances in total.

5.2 Baselines

We compare our model with state-of-the-art baselines: 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) is a heterogeneous GNN baseline that aggregates the investment network via meta-paths and propagates labels over the resulting graph to predict startup outcomes. **GST** (Lyu et al., 2025) is a temporal graph baseline that 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 **BERT Fusion** (Maarouf et al., 2025) concatenates BERT embeddings of each startup's Crunchbase[4] self-description with structured fundamentals and trains a lightweight neural classifier to predict success. **Text RAG** (Lewis et al., 2020) retrieves top-k similar company/investor documents from a local corpus using a dense retriever and conditions a single LLM on the retrieved context. **SSFF** (Wang et al., 2025) is an LLM-based investment-scoring system combining multi-agent analysis, a lightweight predictor, and an external knowledge retrieval module. **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.

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

5.3 Evaluation Metrics

While standard binary classification metrics (precision/recall/F1) are informative, they do not directly capture the VC workflow of selecting a small set of top candidates. We therefore report Precision@K (P@K), the fraction of successful companies among the top-K recommendations ranked by predicted confidence, a common choice in VC prediction (Sharchilev et al., 2018; Zhang et al., 2021; Lyu et al., 2021). To measure robustness over time, we further compute monthly Average Precision@K (AP@K) and report its average across monthly cohorts; higher AP@K indicates more consistent prioritization of successful startups.

5.4 Parameter Settings

We fine-tune Qwen3-4B (Yang et al., 2025) as the base model for our manager. All training runs are conducted on 2× NVIDIA RTX 4090 GPUs (24GB each). We use Sentence-BERT to embed textual inputs. Additional details are provided in the Appendix.

5.5 Model Performance

All reported metrics are averaged over five independent runs of the LLM. Table 1 shows that MIRAGE-VC improves AP@5, AP@10, and AP@20 by **+22.1%**, **+9.3%**, and **+0.3%** (relative to the strongest baselines), respectively—metrics that directly reflect ranking quality when only a handful of top candidates can be pursued in practice. Notably, the gain is largest at small K (AP@5), indicating that higher-confidence selections are substantially more accurate, which is especially valuable for real-world investment screening. It also achieves a **+5.9%** relative gain in F1 and a **+4.2%** relative gain in Precision over the best competing methods, indicating a more accurate and reliable

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	AUC-ROC
SHGMNN	25.41	24.56	26.22	20.65	82.37	32.97	54.1
GST	26.71	25.71	27.14	21.75	**83.54**	34.51	55.6
BERT Fusion	24.67	26.67	25.33	23.63	24.95	24.27	54.3
Text RAG	24.43	24.12	25.23	23.12	60.34	33.43	56.2
SSFF	28.23	30.02	28.42	23.23	69.41	34.81	57.1
GNN-RAG	29.42	27.53	27.04	22.81	71.10	34.54	57.4
Ours	**35.92**	**32.82**	**28.51**	**24.62**	73.30	**36.86**	**61.59**

identification of successful outcomes. Compared to GNN-based methods, which often trade precision for broad structural coverage, and prior LLM/RAG predictors, which may surface redundant evidence and miss high-value multi-hop relations, MIRAGE-VC better filters low-utility graph evidence and surfaces top-performing investment candidates.

Results on the public Crunchbase data are consistent: **+5.5%** AP@5, **+1.44%** F1, and **+3.93%** AUC-ROC over the strongest baseline. See Appendix A.4 for details.

5.6 VC-Expert Evaluation of Rationale

We further assess interpretability with a senior VC expert on 20 held-out companies using a blinded pairwise comparison (Ours vs. an untrained base). The expert assigns two binary labels: Professional (VC-style decision quality) and No critical factual error (no evidence-breaking issue), and selects the overall better rationale. The expert prefers Ours in 85% of cases and rates it as Professional in 100% (vs. 70% for the base), with no critical factual issues (100% vs. 90%). These confirm our model produces **professional, factually grounded investment rationales**. Details of the questionnaires and results are in Appendix A.2.

6 Ablation Study

Table 2 summarizes the ablations. In evidence construction, removing subgraph retrieval yields the largest drop (AP@5: -6.82%, Precision: -1.17%), while replacing IG with random selection also degrades performance (AP@5: -4.32%, Precision: -0.57%); removing external VC knowledge further hurts (AP@5: -2.12%, Precision: -0.47%). For training objectives, SFT-only underperforms preference optimization (AP@5: -2.92%, Precision: -0.77%); DPO improves over SFT (AP@5: $+1.50\%$, Precision: $+0.20\%$), and noise-aware DPO performs best, surpassing standard DPO by $+1.42\%$ AP@5 and $+0.57\%$ Precision.

Table 2: Ablation study (%). Absolute drops are computed against the full model (SFT + noise-aware DPO).

Ablation	AP@5	Precision
(I) Evidence construction		
w/o Subgraph Retrieval	29.10 (↓6.82)	23.45 (↓1.17)
w/o IG Scoring (random)	31.60 (↓4.32)	24.05 (↓0.57)
w/o VC Knowledge	33.80 (↓2.12)	24.15 (↓0.47)
(II) Training objective		
SFT only	33.00 (↓2.92)	23.85 (↓0.77)
SFT + DPO	34.50 (↓1.42)	24.05 (↓0.57)
SFT + **noise-aware DPO**	**35.92** (↓0.00)	**24.62** (↓0.00)

7 Conclusion

We propose MIRAGE-VC, an evidence-grounded reasoning framework for off-graph VC outcome prediction. MIRAGE-VC distills large investment networks into compact, high-utility evidence subgraphs via information-gain signals, and augments reasoning with a two-layer VC knowledge base (theory and practice) to produce explicit, human-auditable rationales. To address one-sided label noise in private-market outcomes, we fine-tune a compact manager via SFT followed by noise-aware DPO, where a PU-based unreliability estimator down-weights updates from likely latent-success negatives. Evaluations on both proprietary and public datasets demonstrate consistent state-of-the-art performance. VC-expert review further shows that MIRAGE-VC generates more professional and internally consistent decision rationales than baselines. Our marginal-utility evidence selection paradigm generalizes to other off-graph prediction tasks where graphs serve as structured evidence; we release code and model outputs to support reproducibility.

Limitations

Geographic Coverage. Our datasets primarily include startups from the US and Europe, with limited representation from major Asian and Latin American markets. Investment patterns, regulatory environments, and entrepreneurial cultures vary significantly across regions. Future work should validate our approach on geographically diverse datasets, particularly from China, India, and Southeast Asia, to ensure cross-market generalizability.

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

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.

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

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.

Benoît Frénay and Michel Verleysen. 2013. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869.

Pierre Giot and Armin Schwienbacher. 2007. Ipos, trade sales and liquidations: Modelling venture capital exits using survival analysis. *Journal of Banking & Finance*, 31(3):679–702.

Paul A Gompers. 2022. Optimal investment, monitoring, and the staging of venture capital. In *Venture capital*, pages 285–313. Routledge.

Paul A Gompers, Will Gornall, Steven N Kaplan, and Ilya A Strebulaev. 2020. How do venture capitalists make decisions? *Journal of Financial Economics*, 135(1):169–190.

Yael V Hochberg, Alexander Ljungqvist, and Yang Lu. 2007. Whom you know matters: Venture capital networks and investment performance. *The journal of finance*, 62(1):251–301.

David H Hsu. 2004. What do entrepreneurs pay for venture capital affiliation? *The journal of finance*, 59(4):1805–1844.

Steven N Kaplan and Josh Lerner. 2016. Venture capital data: Opportunities and challenges. *Measuring entrepreneurial businesses: Current knowledge and challenges*, pages 413–431.

Steven N Kaplan and Per ER Strömberg. 2004. Characteristics, contracts, and actions: Evidence from venture capitalist analyses. *The journal of finance*, 59(5):2177–2210.

David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146.

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

Moonchul Kim and Jay R Ritter. 1999. Valuing ipos. *Journal of financial economics*, 53(3):409–437.

Ryuichi Kiryo, Gang Niu, Marthinus C Du Plessis, and Masashi Sugiyama. 2017. Positive-unlabeled learning with non-negative risk estimator. *Advances in neural information processing systems*, 30.

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, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

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*.

Zhongyang Liu, Haoyu Pei, Xiangyi Xiao, Xiaocong Du, Yihui Li, Suting Hong, Kunpeng Zhang, and Haipeng Zhang. 2025. Llm agents as vc investors: Predicting startup success via roleplay-based collective simulation. *arXiv preprint arXiv:2512.22608*.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2023. Reasoning on graphs: Faithful and interpretable large language model reasoning. *arXiv preprint arXiv:2310.01061*.

Shiwei Lyu, Xiaofeng Li, Suting Hong, Qing Ke, Jinjie Gu, Kunpeng Zhang, and Haipeng Zhang. 2025. Help me screen: Analyzing and predicting the success of start-ups in dynamic venture capital networks. *ACM Transactions on Intelligent Systems and Technology*.

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.

Ian C MacMillan, Robin Siegel, and PN Subba Narasimha. 1985. Criteria used by venture capitalists to evaluate new venture proposals. *Journal of Business venturing*, 1(1):119–128.

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*.

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.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*.

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.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2022. Learning from noisy labels with deep neural networks: A survey. *IEEE transactions on neural networks and learning systems*, 34(11):8135–8153.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M Ni, Heung-Yeung Shum, and Jian Guo. 2023. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. *arXiv preprint arXiv:2307.07697*.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2023. Can language models solve graph problems in natural language? *Advances in Neural Information Processing Systems*, 36:30840–30861.

Xisen Wang, Yigit Ihlamur, and Fuat Alican. 2025. 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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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.

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*.

Bob Zider. 1998. How venture capital works. *Harvard business review*, 76(6):131–139.

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., 2025), SimVC-CAS (Liu et al., 2025), and FinCon (Yu et al., 2024) systems establish analyst collaboration mechanisms, 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.

A.2 Interpretability of Manager Reasoning Outputs

As illustrated in Figure 4, we evaluate the interpretability and professionalism of the manager's reasoning outputs via a VC expert review on 20 held-out companies. The expert scores the quality of the written rationale (not whether the prediction is correct). For each company, two anonymized outputs(from 2 models) are presented in randomized order, and the expert (i) labels each output and (ii) selects an overall best one.

Base is the unmodified instruction-tuned backbone (Qwen3-4B-Instruct without task-specific fine-tuning). **Ours** is our final manager after task-specific training (SFT followed by our preference-based optimization), used as the decision model in all main experiments.

Evaluation rubric and criteria. The expert provides binary labels for each output: **Professional** (whether the reasoning reads like a VC decision memo with clear, coherent synthesis) and **No critical factual error** (whether there exists any evidence-breaking factual issue that would undermine credibility). In addition, the expert selects the **best overall** rationale between the two outputs for each company.

Results and analysis. Table 3 shows that **Ours** produces substantially more professional rationales (**100%** vs. **70%** for Base) and is preferred in the majority of pairwise comparisons (**85%** vs. **15%**). This suggests that our training procedure improves not only the stylistic polish but also the decision coherence of the final synthesis, making the reasoning more consistent with how VC analysts write internal memos. Regarding factual reliability, both models are largely faithful to the provided evidence, while **Ours** further reduces critical factual issues (**100%** vs. **90%**). Overall, these results support the claim that our manager produces more interpretable and trustworthy rationales suitable for downstream human review and auditing.

Metric	Base	DPO
Professional	70.0	100.0
Best overall win	15.0	85.0
No critical factual error	90.0	100.0

Table 3: VC expert evaluation (%).

A.3 External VC Knowledge Base: Data Sources and Curation

This appendix describes the construction of our two-layer VC knowledge base and its curation protocol.

Two-layer organization. We structure the library into two complementary layers: (i) **VC theory**, which captures stable and broadly applicable principles (e.g., staging, monitoring, contracting, selection), and (ii) **VC practice**, which summarizes operational screening and diligence heuristics (e.g., evaluation criteria, decision workflows, practitioner playbooks).

Candidate sources. Candidates are collected from two streams: (i) academic books, surveys, and peer-reviewed papers on VC mechanisms and outcomes; and (ii) practitioner-oriented materials that explicitly articulate screening and diligence processes.

Unified inclusion criterion. A document is retained only if it satisfies both (a) **topical relevance** to VC decision-making/mechanisms and (b) a **quality threshold** intended to proxy community validation. Concretely, we retain a candidate if it is either:

- An academic source with at least **100** citations;

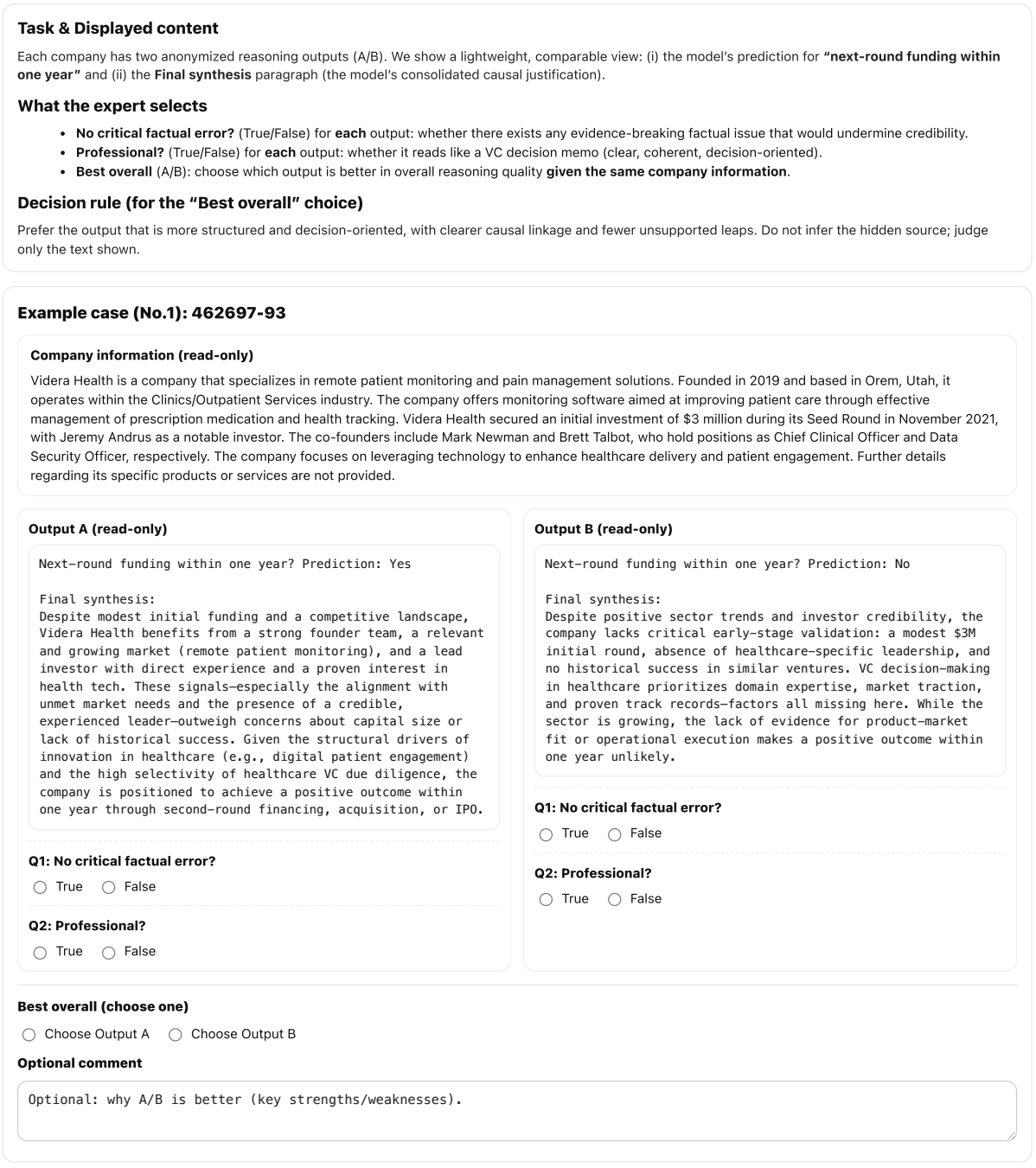

Figure 4: Example of the VC expert evaluation interface and a scored case.

- A practitioner source that is **corroborated by multiple independent** VC playbooks and frameworks (e.g., repeatedly cited or referenced across distinct practitioner documents).

Layer assignment. Within the retained set, each document is assigned to the layer that best reflects its primary focus: theory sources emphasize general principles and causal mechanisms, whereas practice sources emphasize actionable screening criteria and decision procedures.

Representative sources.

- **VC theory:** *Optimal investment, monitoring, and the staging of venture capital* (Gompers, 2022); *How venture capital works* (Zider, 1998).

- **VC practice:** *Criteria used by venture capitalists to evaluate new venture proposals* MacMillan et al. (1985); *How do venture capitalists make decisions?* (Gompers et al., 2020).

A.4 Crunchbase Public-Dataset Evaluation

Dataset. We use the public Crunchbase snapshot (2013, 196,553 companies, 226,708 investors/people) to validate reproducibility. Following the main setting, we construct a time-stamped investor–company investment graph with directed edges (investor $\rightarrow$ company) annotated by investment time (and optionally round/amount when available).

Task and labels. We adopt the same task definition as in the main experiments: given a company that completes its seed/angel (or first) financing at time t_0, predict whether it will secure the next-round funding (Series A) within 12 months. We set $y = 1$ if such an event occurs within $[t_0, t_0{+}12$ months$]$, and $y = 0$ otherwise.

Splits. We apply a temporally ordered split with an overall train:val:test ratio of approximately 7:1:2. Concretely, Crunchbase contains 4,375 labeled instances in total, where train/val precede the test period to avoid overlap.

Training and baselines. All model components, prompts, and evaluation protocols follow the main experiments. We compare MIRAGE-VC against the same baselines (GNN-based, text+tabular, and RAG-based LLM baselines) under identical metric definitions.

Method	AP@5	F1	AUC-ROC
SHGMNN	24.90	29.10	52.60
GST	26.80	30.50	54.10
BERT Fusion	25.10	31.60	55.80
Text RAG	25.80	30.60	53.50
SSFF	26.10	31.05	55.80
GNN-RAG	28.05	30.70	56.85
MIRAGE-VC (Ours)	**33.54**	**33.04**	**59.73**

Table 4: Performance on public Crunchbase data. Values are percentages (% omitted).

Results. Table 4 reports results on Crunchbase. MIRAGE-VC achieves consistent improvements over the strongest baseline: **+5.5%** AP@5, **+1.44%** F1, and **+3.93%** AUC-ROC, matching the trends observed on PitchBook.

A.5 Analysis-Agent Backbone Sensitivity

To examine how sensitive MIRAGE-VC is to the backbone LLMs used by the three perspective analysis agents (peer-company, lead-investor, and graph-evidence analysts), we reran the full pipeline while only swapping the analysis-agent backbones among GPT-3.5-Turbo, GPT-4o-mini, and Qwen-3 4B. All other components are held fixed, including the retriever/selector, external knowledge retrieval, and the same trained manager with an identical memo schema. Thus, the overall input/output interface remains unchanged; only the intermediate analytical text produced by the three agents differs.

As shown in Table 5, MIRAGE-VC exhibits limited sensitivity to the backbone LLMs used by the three analysis agents. Across backbones, all metrics remain within a narrow band, indicating that the downstream manager can robustly utilize the agents' intermediate analyses despite variations in generation style and capability. We observe only marginal metric-wise trade-offs: GPT-3.5-TURBO performs best on AP@5/AP@10 and F1, whereas GPT-4O-MINI yields slight gains on longer-horizon retrieval (AP@20) and ranking quality (AUC-ROC). Overall, these results suggest that MIRAGE-VC does not depend on any single proprietary LLM for the analysis agents, and its end-to-end performance is largely preserved under backbone substitutions.

A.6 Explanation of low recall

As outlined in the paper and supported by prior VC research, real-world investment scenarios prioritize identifying the best few companies under limited resources, making AP@k the most relevant metric for evaluation. In this regard, our method consis-

Table 5: Sensitivity to the backbone LLMs used by the three analysis agents (peer-company, lead-investor, and graph-evidence analysts).

Analysis-agent backbone	AP@5	AP@10	AP@20	Precision	Recall	F1	AUC-ROC
GPT-3.5-Turbo (default)	**35.92**	**32.82**	28.51	**24.62**	73.30	**36.86**	61.59
GPT-4o-mini	35.63	32.52	**29.40**	24.30	**73.41**	36.51	**61.71**
Qwen-3 4B	34.60	31.60	28.00	24.00	72.60	36.00	60.60

tently achieves significantly higher AP@k scores than all baselines, which reflects its effectiveness for practical application.

Regarding recall, it is important to note that the two methods with the highest recall are early GNN-based approaches. These methods achieve higher recall primarily due to(potential):

- **Over-reliance on neighborhood aggregation:** These models aggregate broad neighborhood information, which increases recall but also introduces noise.

In contrast, our proposed method is designed to optimize the precision–recall trade-off and achieves the best overall performance across composite metrics. This aligns with the practical requirements of scenarios like VC prediction, where precision in selecting high-potential targets is far more critical than simply achieving high recall.

A.7 More Details about Data

A.7.1 Data Sources and Time Filtering of data

Our entity-level documents are built by concatenating descriptive tabular fields (e.g., team members, educational background, sector, stage, region) and explicitly exclude outcome/status columns tied to the prediction targets (acquisition, IPO, financing); combined with time filtering that restricts text to information available before the prediction cutoff, this design minimizes label-leakage risk. As an empirical check, we keyword-searched a random sample of 200 company profiles and 300 investor profiles for `acquisition`, `IPO`, and `financing` and found no evidence of direct leakage. We will include implementation details of the document construction and time-filtering procedures in the revised paper.

A.7.2 Interaction Between Graph and Auxiliary evidence Retrieval

As mentioned above, our raw data originate from relational tables. For each entity, we build a textual profile by concatenating salient columns, followed by normalization and deduplication. This yields a corpus of entity-level documents used across modules.

Auxiliary evidence retrieval. Auxiliary evidence retrieval operates directly over these constructed textual profiles, selecting semantically relevant entities from the corpus (e.g., via dense or keyword retrieval).

Graph evidence retrieval. From the relational tables we induce an investment network and retrieve key paths based on relations (e.g., company–investor–portfolio links). The nodes along the selected paths are then *mapped back* to their textual profiles, which are consumed by downstream modules.

A.8 Implementation Details

A.8.1 Graph evidence 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 $w = (t_1, \ldots, t_m)$, 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 Hyperparameter is listed in Table 7.

Evaluation and Results With the introduction of the STOP option, the selector is evaluated not only on *within-group ranking quality* but also on its ability to *decide whether to terminate*. At each step, the candidate set consists of frontier nodes augmented with STOP, where STOP is assigned a fixed gain of 0 and thus serves as the natural boundary between beneficial and non-beneficial expansions. We report three complementary metrics.

NDCG@1 measures top-1 ranking quality by normalizing the gain of the selected action against the maximum attainable gain within the candidate group; it assigns partial credit when the chosen action is near-optimal and directly reflects how well the selector prioritizes high-gain expansions.

StopHit (AllNeg) is computed on groups where no frontier candidate yields positive gain (i.e., STOP is the oracle-optimal action); it measures the selector's ability to correctly terminate when further expansion is unhelpful.

NonStopHit (HasPos) is computed on groups that contain at least one positive-gain candidate; it measures the selector's ability to avoid premature termination and continue exploring when beneficial expansions exist.

A random baseline assigns i.i.d. scores sampled from $\mathcal{U}(0,1)$ to all actions (including STOP) and applies the same evaluation protocol. As shown in Table 6, the selector consistently outperforms random across all metrics, indicating that it learns both (i) to rank candidates by task-specific gain, and (ii) to make appropriate termination decisions via STOP.

A random baseline assigns i.i.d. scores sampled from $\mathcal{U}(0,1)$ to all actions (including STOP) and applies the same evaluation protocol. As shown in Table 6, the selector consistently outperforms random across all metrics, indicating that it learns both (i) to rank candidates by task-specific gain, and (ii) to make appropriate termination decisions via STOP.

Table 6: Subgraph selector performance.

Method	NDCG@1	StopHit (AllNeg)	NonStopHit (HasPos)
Random $\mathcal{U}(0,1)$	0.504	0.252	0.748
SUBGRAPH SELECTOR	**0.594**	**0.418**	**0.785**

Table 7: Hyper-parameters for the Subgraph Selector

Parameter	Subgraph Selector
Text vector dimension	384
Batch size	256
Training epochs	30
Hidden width	256
Optimiser	AdamW
Learning rate	3×10^{-4}
Temperature τ	0.5

A.8.2 Improved PU Calibration for One-Sided Noise

This appendix complements §4.4.2 by specifying (i) how we implement the instance-dependent propensity $\hat{c}(x)$ as a piecewise function, (ii) how the Precision@K anchor is used for calibration, and (iii) the practical hyperparameters and calibration statistics used to compute $q(x)$ for weighting.

OOF scoring for stable $\hat{g}(x)$. We train a PU classifier $g(x) \approx \Pr(s = 1 \mid x)$ (with $s = 1$ for observed positives and $s = 0$ for unlabeled) and compute out-of-fold (OOF) predictions $\hat{g}(x)$ using 5-fold stratified splits. We use OOF scores in calibration to reduce overfitting and avoid overly optimistic $g(x)$ that would destabilize $q(x)$.

Piecewise $\hat{c}(x)$ over simple strata. As stated in Eq. (11)–(12), we model the recording propensity as instance-dependent $c(x) = \Pr(s = 1 \mid y = 1, x)$. To keep calibration lightweight, we implement $\hat{c}(x)$ as a piecewise constant function over coarse instance strata,

$$\hat{c}(x) = \hat{c}_{b(x)}, \quad b(x) \in \{1, \ldots, B\}, \tag{16}$$

where $b(x)$ is a bucket index (e.g., sector/stage/region). Each bucket is calibrated independently on a calibration split using the same high-confidence anchor described below. In practice, this captures major heterogeneity in data coverage while avoiding a heavy parametric model for $c(x)$.

Precision@K anchor and calibration objective. On the calibration split, let $H_{K,b}$ denote the top-K *high-confidence* instances *within bucket* b, ranked by a fixed base score (we use the SFT manager's implied success probability; this calibration is done *before* noise-aware DPO to avoid circularity). We compute the observable anchor

$$\widehat{\mathrm{P@K}}_b = \frac{1}{|H_{K,b}|} \sum_{x \in H_{K,b}} \mathbb{I}\big[y_{\text{obs}}(x) = 1\big] \tag{17}$$

$$\hat{r}_b = 1 - \widehat{\mathrm{P@K}}_b \tag{18}$$

where $\hat{r}_b$ reflects the observed-negative fraction in a region that the model considers highly likely to be positive, and thus serves as a direct signal of latent-success contamination among $y_{\text{obs}} = 0$

We then calibrate the bucket-level propensity $\hat{c}_b$ by a 1D search so that the *implied* contamination among observed negatives in the same high-confidence region matches $\hat{r}_b$ Specifically, for a candidate $\tilde{c} \in (0, 1]$, we compute $q_{\tilde{c}}(x) = \min(1, \hat{g}(x)/\tilde{c})$ for x with $y_{\text{obs}} = 0$, and define the implied contamination estimate in bucket b as

$$N_{0,b} \;=\; \big|\{x \in H_{K,b} : y_{\text{obs}}(x) = 0\}\big| \tag{19}$$

$$\widehat{r}_b(\tilde{c}) = \frac{1}{N_{0,b}} \sum_{\substack{x \in H_{K,b} \\ y_{\text{obs}}(x)=0}} q_{\tilde{c}}(x) \tag{20}$$

We choose $\hat{c}_b$ by minimizing the absolute mismatch:

$$\hat{c}_b \;=\; \arg\min_{\tilde{c} \in (0,1]} \left| \widehat{r}_b(\tilde{c}) - \hat{r}_b \right| \tag{21}$$

implemented via a coarse-to-fine grid search (details below). Finally, we set $\hat{c}(x) = \hat{c}_{b(x)}$.

$q(x)$ computation and stabilization. For observed negatives, we compute

$$q(x) = \mathbb{I}[y_{\text{obs}} = 0] \cdot \min\left(1, \frac{\hat{g}(x)}{\hat{c}(x)}\right) \tag{22}$$

and set $q(x) = 0$ for observed positives. To prevent saturation and improve stability, we apply (i) clipping to $[0, 1]$, (ii) a mild shape compression on observed negatives, $q(x) \leftarrow \max(q(x), \epsilon)^{\gamma}$, and (iii) a lower bound $w_{\min}$ in Eq. 13.

Hyperparameters and representative calibration statistics. We implement $g(x)$ using TF-IDF features over the manager prompt (max_features=200K, ngram=(1,2), min_df=2) and logistic regression ($C = 2.0$, max_iter=2000), with 5-fold OOF. For stabilization we use $\gamma = 2.5$ and $\epsilon = 10^{-6}$; K is chosen to match the main Precision@K operating point in §*Experiments*. Table 8 reports representative calibration statistics from our main setting, including proxy AUC for separating $s = 1$ vs. $s = 0$ (sanity check) and the resulting $q(x)$ saturation rate on observed negatives.

Statistic	Value
#instances (n)	8,904
#observed positives / negatives	2,493 / 6,411
Proxy AUC for s (val)	0.629
OOF folds	5
Shape compression γ	2.5
Saturation frac. $\Pr[q(x) = 1 \mid y_{\text{obs}} = 0]$	0.000

Table 8: Representative PU calibration statistics (main setting).

A.9 Greedy Evidence Construction vs. Global Optimality

Our evidence construction uses a greedy marginal-utility criterion (information-gain style Δ), which is inherently local and may not match the *globally optimal* subgraph under a fixed hop/token budget. However, the globally optimal objective here is a combinatorial subset-selection problem: among all feasible subgraphs within the budget, choose the one that maximizes downstream utility. This is closely analogous to classic influence maximization on graphs, which is NP-hard, and thus exact global optimization is computationally infeasible at our graph scale (Kempe et al., 2003). Therefore, adopting greedy marginal-gain selection is a standard and practical approximation in large-scale graph settings, providing an effective substitute for intractable global-optimal selection.

A.10 Text Embedding Model Analysis

One pipeline component relies on a frozen sentence encoder to obtain text representations: the *subgraph selector*, where the encoder embeds the verbalized candidate actions (baseline vs. expanded subgraph) to form selector features. Here we analyze whether swapping the encoder affects the selector's ranking quality. Table 9 shows that the NDCG@1 scores are highly consistent across alternative encoders, indicating that our retrieval performance is not sensitive to the specific choice of text-embedding model.

Table 9: Impact of text encoders on the SUBGRAPH SELECTOR.

Text encoder	Dim.	NDCG@1
all-MiniLM-L6-v2	384	59.4
jina-embeddings-v2-base	768	59.6
e5-large-v2	1024	59.1

A.11 Resource Utilization and Latency

Our end-to-end pipeline comprises three phases: (i) evidence construction with three GPT-3.5 API calls

per instance, (ii) oracle gain annotation with Llama-3.1-8B and subgraph-selector training, and (iii) local manager LLM training (SFT and noise-aware DPO). Overall, we issued approximately ∼30,000 GPT-3.5 requests (under 12,000 tokens each), processing ≈360 million tokens. 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 subgraph selector on an NVIDIA RTX 4090 (24 GB VRAM) required only a few minutes, while training the manager LLM with SFT and DPO took approximately ∼20 GPU hours. Including hyperparameter sweeps and auxiliary runs, the total GPU budget is on the order of a few dozen GPU hours.

During inference, we noticed that LLM usage is no higher than other well-known LLM-based forecasting systems, such as SSFF and GNN-RAG. This difference in cost is particularly insignificant for VC investment forecasting, a non-immediate and low-frequency task (weekly or monthly) that differs substantially from stock price forecasting. In our process, each prediction proceeds in two stages: the first stage issues three API calls to produce the multi-view analyses, and the second stage runs a *local* manager LLM to output the final decision memo. On average, a single prediction consumes $\sim$ 14,100 API input tokens and takes $\sim$ 6.5 s wall-clock end-to-end, where the local manager introduces only minor additional latency. This per-instance usage is operationally acceptable. In a practical setting with 1,000 new target companies per month, the theoretical API-side inference budget is $\sim$ 14.1 million input tokens in total, costing roughly $\sim$ \$1.4 with GPT-4o-mini (API-side only), and requiring about $\sim$ 1.8 h end-to-end runtime, which is operationally acceptable.

A.12 Prompt Templates and Examples

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

A.12.1 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\\
$\bullet$ Senior Engineer, ABB Robotics (2008 --
     2012)---global industrial-robotics leader
    .\{COMPANY\_PROFILE\} (success)\\
$\bullet$ Investment Associate, TechEdge Capital
     (2012 -- 2016)---early-stage deep-tech VC
    .\{COMPANY\_PROFILE\} (success)

Focus sectors    : Robotics $\bullet$ Edge AI $\
    bullet$ IoT\\
Assets under mgmt: USD 1.4 B

Investment record\\
$\bullet$ RoboVac: acquired by Dyson (2021). \{
    COMPANY\_PROFILE\} (success)\\
$\bullet$ MechArm: IPO (2022). \{COMPANY\
    _PROFILE\} (success)\\
$\bullet$ NanoGrip: acquired by Bosch (2020). \{
    COMPANY\_PROFILE\} (success)\\
$\bullet$ ServoLink: ceased operations (2019).
    \{COMPANY\_PROFILE\} (failure)

Board seats : MechArm $\bullet$ FlexDroid $\
    bullet$ SensorX\\
Awards      : Forbes ``30 Under 40 in VC''
    (2023)
```

A.12.2 Subgraph 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.12.3 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.12.4 Investor Analyst Prompt

```
Role:
  You are a senior venture-capital analyst who
    specialises in evaluating a
  start-up's 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 r\'esum\'e (prior operating
    roles and portfolio companies,
    each annotated as success or failure---
    success = the company raised Series A
    within 12 months of its seed/angel round;
    failure = it did not):\{INVESTOR\_PROFILE\}

Task:
  $\bullet$ Analyse how the investor's past
    successes and failures relate to the target
    company's sector, stage, and needs.
  $\bullet$ Predict whether the target will
    raise a Series-A round within 12 months.
  $\bullet$ Output \textbf{exactly} in the
    format:

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

  $\bullet$ If evidence is insufficient, reason
    cautiously but still decide.
```

A.12.5 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
    $\bullet$ Prediction: \{PATH\_PREDICTION\}
    $\bullet$ Analysis  : \{PATH\_ANALYSIS\}

(2) Similar-company analyst verdict
    $\bullet$ Prediction: \{SIM\_PREDICTION\}
    $\bullet$ Analysis  : \{SIM\_ANALYSIS\}

(3) Lead-investor analyst verdict
    $\bullet$ Prediction: \{INV\_PREDICTION\}
    $\bullet$ 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:
  $\bullet$ Produce a single, final prediction
    on whether the target will raise a
    Series-A round within 12 months.
  $\bullet$ Output \textbf{exactly} in the
    format:

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

  $\bullet$ If evidence is insufficient, reason
    cautiously but still decide.
```