Which campus does this invoice belong to?

Backtesting invoice classification on the billing@alpha.school mailbox against QuickBooks · Initiative B · September 28, 2026

84.0%	91.4%	34%	958
best method (routing ensemble), always answering, September	accuracy on the 79% it answers at ≥ 0.9 confidence	Jev as the classifier (75 labels), always answering	September messages scored, 15 companies

> **Bottom line.** Using only what is knowable when an invoice email arrives, a **routing ensemble** — a gradient-boosting model over email, context and LLM-extracted features, overridden by the LLM when the LLM confidently names a specific non-HQ campus — assigns the correct QuickBooks company to **84% of September's invoice emails** (95% CI 82–86%) among 15 companies, and to **91% of the 79%** it is most confident about. Asking an LLM directly gets 44% (it abstains on half and is 85% right on the rest); using Jev directly as the classifier gets 34–39%. The learned models win because they see what the email doesn't say: who this sender and this vendor have billed before.

Three results matter most for building this:

1. **Context beats content, and content beats nothing.** The biggest single lever is QuickBooks history *as of arrival* — which companies a sender and vendor have billed before — which only a system with QuickBooks access has. Zero-shot models (LLM, Jev) never see it and fall to ~25% on follow-ups and ~20% on Austin K-8.
2. **The zero-shot LLM and the learned model fail in opposite places**, which is why routing between them wins: the learned model is right about the headquarters campus and about follow-ups; the LLM is right about small and newer campuses that the learned model absorbs into Austin.
3. **Alpha's headquarters address is the main source of error.** It is printed as the Bill-To on invoices booked to many campuses and shares a ZIP with Austin K-8; hand-written rules and the LLM both read it as "Austin K-8".

---

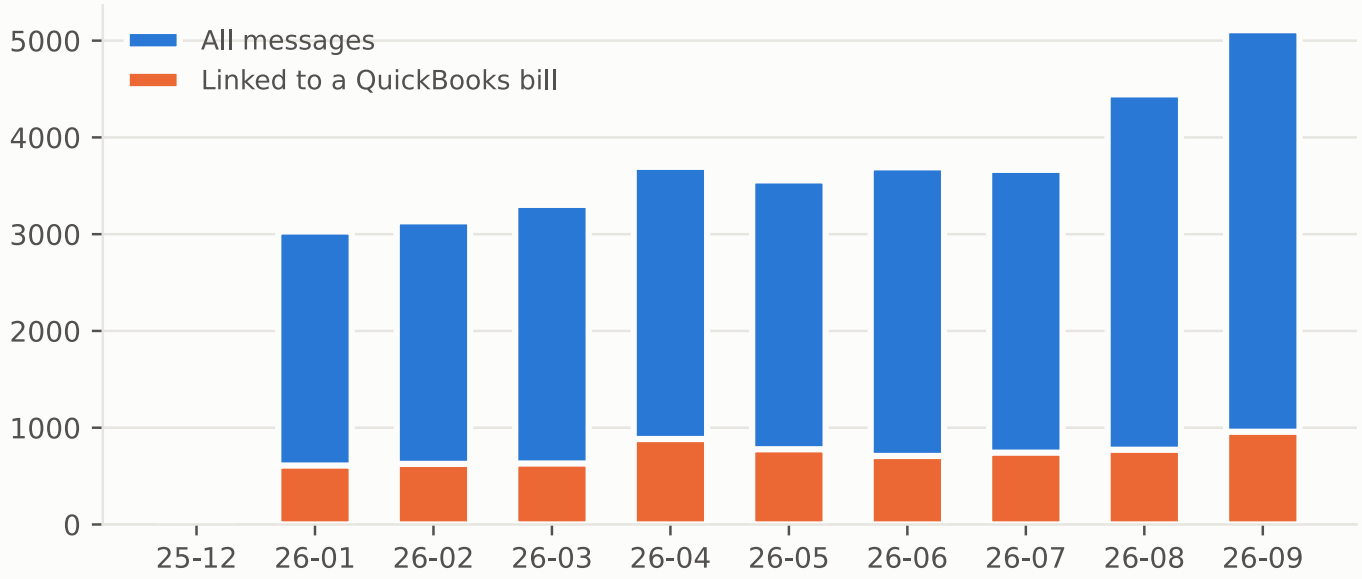
1. The question and the data

Every invoice Alpha School pays reaches **billing@alpha.school** by email, and every bill it books lands in one of ~73 QuickBooks companies, one per legal entity (campus). The task: **given an email as it arrives, predict the company its bill will be booked to** — or say it can't tell.

- **Emails.** A read-only export of the mailbox's `All Mail` from 1 January to 27 September 2026, with headers, bodies, attachments, Gmail thread ids and the AP team's labels. Read/unread state was never touched.
- **Ground truth.** Read-only access to 16 QuickBooks companies (15 with bills), including every bill, its vendor, amount, account, class and creation time. An email is *linked* to a bill when there is strong evidence: a byte-

identical or same-named attachment, the vendor's invoice number in the text, or the vendor name plus the exact amount. The linked emails are the scored population.

- **Split, by time.** January–July = training, August = validation (every design choice made here), September = test, scored once with the design frozen and the models refit on January–August.



Messages per month in the archive, and those linked to a QuickBooks bill (the scored population).

2. Guarding against leakage

A prediction may only use what existed **when the email arrived**. That rule was enforced mechanically, and an audit partway through found and fixed two violations:

- **Features are strictly as-of.** Every history feature counts only QuickBooks bills whose *creation time* precedes the email's arrival (with timezone).
- **Thread context was leaking — fixed.** It originally looked at earlier emails in the same thread that had been linked to a bill, even when that bill was entered *after* the current email arrived. Now only bills that already existed count. The leak had inflated the rule-based methods by 4–7 points.
- **No tuning on the test month.** Earlier iterations had adjusted rules and thresholds while looking at September. The protocol now makes every choice on August; September was scored once.
- **Outcomes are never inputs.** The AP team's Gmail labels, the bill itself and anything recorded later are used only as ground truth.
- **Model-derived features are safe by construction.** The LLM and Jev never see labels. The text model's scores for training rows are *out-of-fold*, so the stacked model never learns from a text score produced by a model trained on that same email.
- **Same features on train and test.** LLM-extracted fields are produced by one model with one fixed prompt and company list for every email; the model only uses them when they cover the training and scored rows alike.

3. Features

Three families, all observable at arrival:

1. **The email.** Sender and recipients; whether billing@ was addressed or only copied; origin (sent by the vendor, through an invoicing platform such as QuickBooks or Stripe, or forwarded by an employee — with the original sender recovered from the forwarded block); subject, body, attachment names and the text of PDF attachments.
2. **Derived from the email by models.** An LLM (GLM 5.3 Flash, via the project gateway) extracts what a bookkeeper would: vendor, invoice number, date, amount, Bill-To name/address/ZIP, service location, document type, and its own guess at the company with a confidence. A TF-IDF text model provides a probability per company.
3. **Context as of arrival — the part only a system with QuickBooks has.** For the sender's domain, the forwarded-original domain and the vendor: which companies their earlier bills went to, how many, and how recently; whether the vendor is in each company's vendor list and in how many; which company the thread's earlier bills belong to; each company's recent bill volume.

4. Methods compared

Eleven approaches, from trivial to stacked, all scored on the same September emails:

Approach	What it is	Learns from labels?
Majority	always the most common company	—
Sender history	the company most of this sender's earlier bills went to	no (running count)
Rules	a Bill-To ZIP, any single ZIP, or one campus name in the text	no
Jev, 75 labels	TypeSafe's Jev `Choice` model over all 73 companies + 2 catch-alls	no (zero-shot)
Jev, 16 labels	the same, restricted to the 15 candidate companies + "not campus-specific"	no
LLM zero-shot	GLM 5.3 Flash asked directly, may answer "unknown"	no
Logistic regression	text + tabular features, one-hot per company	yes
Gradient boosting	tabular features only, one-hot per company	yes
Candidate ranker, no LLM	the stacked model below without the LLM features	yes
Candidate ranker	scores every *(email, candidate company)* pair on relationship features and picks the top	yes
Routing ensemble	gradient boosting decides, unless the LLM names a specific non-HQ campus with confidence ≥ 0.9 (cutoff chosen on August)	yes

The candidate ranker is the stacked model. Rather than one-hot features per company, each row is an (email, company) pair described by features such as *this sender's past share of bills to this company*, *the LLM's confidence for this company* and *the out-of-fold text probability for this company*. The features mean the same thing for every company, so one model covers all of them — including campuses that have few or no past bills, which is the situation every time Alpha opens one.

Every method may abstain, so each is reported at three operating points: always answering (abstentions count as wrong), and on the emails it answers with confidence ≥ 0.7 and ≥ 0.9, with the share it answers (coverage).

5. Results (September, scored once)

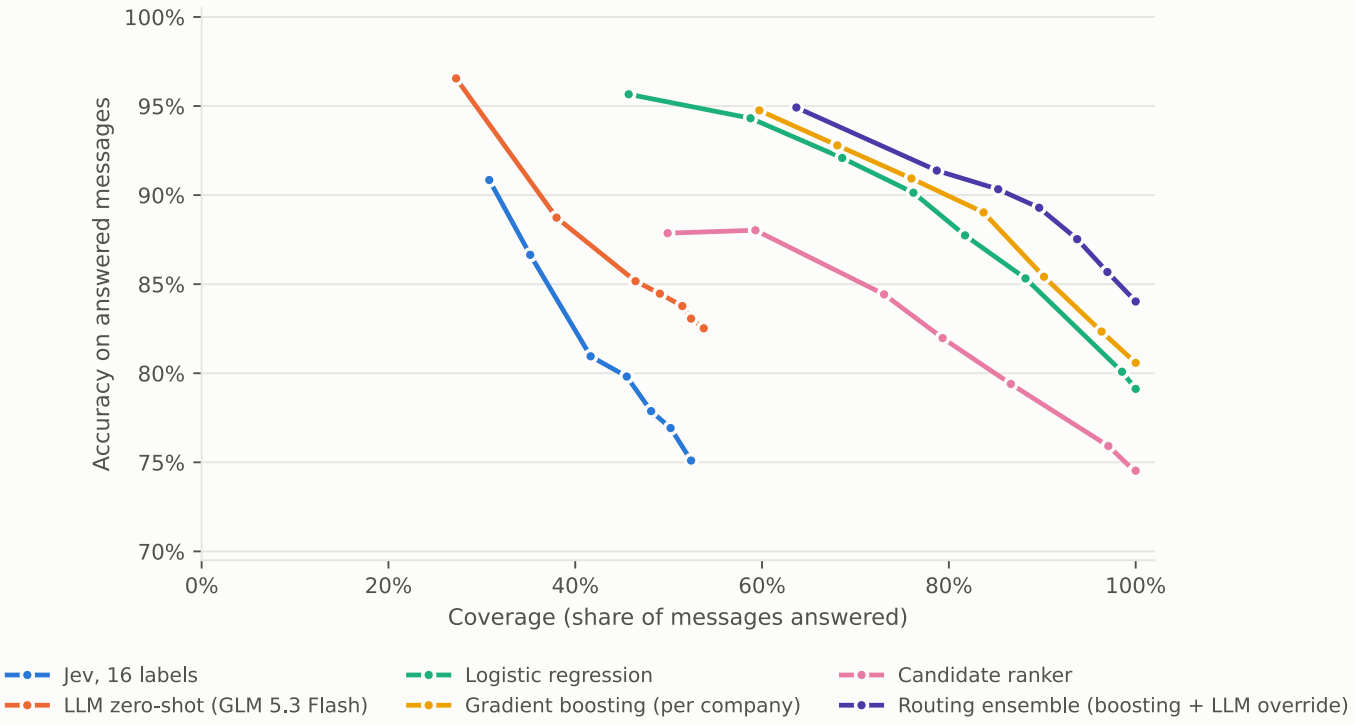
The table scores every method on the same 958 September emails (15 companies; Austin K-8 is 34% of them, so "always Austin" scores 34%). The chart shows the trade-off each method offers between answering more emails and being right more often.

- **Learned models are in a different league from zero-shot ones.** Logistic regression and gradient boosting reach 79–81% always answering and 93–94% on the ~60–70% they answer at ≥ 0.9. The LLM zero-shot is precise when it answers (85% at ≥ 0.7) but answers only half the time; Jev with all 75 labels is barely above the "always Austin" baseline, and restricting it to the same 15 companies lifts it only to 39%.
- **The routing ensemble is best on both months** (77.3% on August, where the rule was fixed; 84.0% on September). It adds nothing on Austin or on follow-ups, where the model is already right, and a lot where the LLM can read the campus: vendor-direct mail 76% → 93%, new invoices 76% → 84%, Chicago 31% → 65%, Marin 0% → 73%.
- **The candidate ranker**, the best method on the earlier six-company data, trails the per-company models here (74.5%). With 15 companies and 5,700 training emails the per-company models have enough examples per class, and the ranker's shared features dilute the company-specific patterns.
- **Rules are worse than useless** once Austin K-8 is a candidate (30%), because the HQ ZIP is Austin's ZIP.

Method	Always answering	Accuracy ≥ 0.7	Coverage	Accuracy ≥ 0.9	Coverage
Majority company	34.4% [32%–38%]	34.4% [32%–38%]	100%	34.4%	100%
Sender history	41.4% [38%–44%]	56.6% [53%–61%]	69%	90.1%	17%
Rules (ZIP / campus name)	30.0% [27%–33%]	50.8% [47%–55%]	59%	50.8%	59%
Jev, 75 labels	34.1% [31%–37%]	56.2% [52%–61%]	51%	61.5%	39%
Jev, 16 labels	39.4% [36%–43%]	79.8% [76%–84%]	46%	86.6%	35%

Method	Always answering	Accuracy ≥ 0.7	Coverage	Accuracy ≥ 0.9	Coverage
LLM zero-shot (GLM 5.3 Flash)	44.4% [41%–48%]	84.5% [81%–88%]	49%	88.7%	38%
Logistic regression	79.1% [77%–82%]	90.1% [88%–92%]	76%	94.3%	59%
Gradient boosting (per company)	80.6% [78%–83%]	89.0% [87%–91%]	84%	92.8%	68%
Candidate ranker, no LLM	71.0% [68%–74%]	80.0% [77%–83%]	82%	85.6%	61%
Candidate ranker	74.5% [72%–77%]	82.0% [79%–84%]	79%	88.0%	59%
Routing ensemble (boosting + LLM override)	84.0% [82%–86%]	89.3% [87%–91%]	90%	91.4%	79%

Brackets: bootstrap 95% confidence intervals. “Always answering” counts abstentions as wrong. Coverage = share of messages the method answers at that confidence.



Coverage vs accuracy as the confidence threshold rises (right to left). A single dot = a method without a usable confidence. Up and to the right is better.

By situation

Slice	n	Jev, 16 labels	LLM zero-shot (GLM 5.3 Flash)	Logistic regression	Gradient boosting (per company)	Routing ensemble (boosting + LLM override)
New invoice	402	61%	68%	75%	76%	84%
Follow-up	556	24%	27%	82%	84%	84%
Employee forward	384	38%	41%	82%	84%	84%
Via invoicing platform	192	51%	59%	76%	80%	87%
Vendor direct	143	63%	69%	82%	76%	93%
Employee thread	143	16%	20%	67%	68%	66%
Vendor reply	92	20%	25%	89%	93%	92%
All other companies	628	50%	57%	73%	73%	82%
Austin K-8	330	19%	20%	90%	95%	88%

Accuracy always answering, per slice. “New invoice”: the bill did not exist in QuickBooks when the email arrived.

By company

Company	n	Jev, 16 labels	LLM zero-shot (GLM 5.3 Flash)	Logistic regression	Gradient boosting (per company)	Routing ensemble (boosting + LLM override)
Alpha School 78746, LLC (Austin K-8)	330	19%	20%	90%	95%	88%
Alpha School 33155, LLC	153	40%	43%	93%	97%	97%
Alpha School 60601, LLC (Chicago)	91	65%	65%	37%	31%	65%
Alpha School 78701, LLC (Alpha High School)	62	32%	48%	92%	89%	90%
Sports Academy 75010, LLC (CSA)	56	43%	64%	89%	89%	89%
Alpha School 00692, LLC (Dorado)	48	67%	69%	79%	92%	92%
Alpha School 94123, LLC (San Francisco)	44	55%	45%	82%	61%	64%
Alpha School 92610, LLC (Orange County)	38	66%	76%	82%	82%	89%

Company	n	Jev, 16 labels	LLM zero-shot (GLM 5.3 Flash)	Logistic regression	Gradient boosting (per company)	Routing ensemble (boosting + LLM override)
Sports Academy 78734, LLC (TSA)	36	44%	75%	67%	75%	75%
Alpha Early Center 10038, LLC (Alpha Early Center Fin District)	28	39%	46%	0%	0%	39%
Alpha School 94306, LLC (Palo Alto)	21	71%	71%	90%	81%	81%
Alpha School 78521, LLC (Brownsville)	19	32%	32%	74%	100%	100%
gt.school 78626, LLC	13	92%	100%	100%	100%	100%
Alpha School 94965, LLC (Alpha Marin)	11	73%	73%	18%	0%	73%
Alpha Anywhere Center 10504, LLC (Alpha Armonk)	8	25%	62%	0%	0%	12%

Validation month (August), for comparison

Method	Always answering	Accuracy ≥ 0.7	Coverage
Majority company	48.2%	48.2%	100%
Sender history	36.1%	54.3%	63%
Rules (ZIP / campus name)	20.6%	36.9%	56%
LLM zero-shot (GLM 5.3 Flash)	36.4%	77.1%	43%
Logistic regression	74.1%	82.8%	80%
Gradient boosting (per company)	73.4%	79.4%	86%
Candidate ranker, no LLM	75.8%	78.3%	90%
Candidate ranker	76.8%	79.7%	89%
Routing ensemble (boosting + LLM override)	77.3%	82.5%	89%

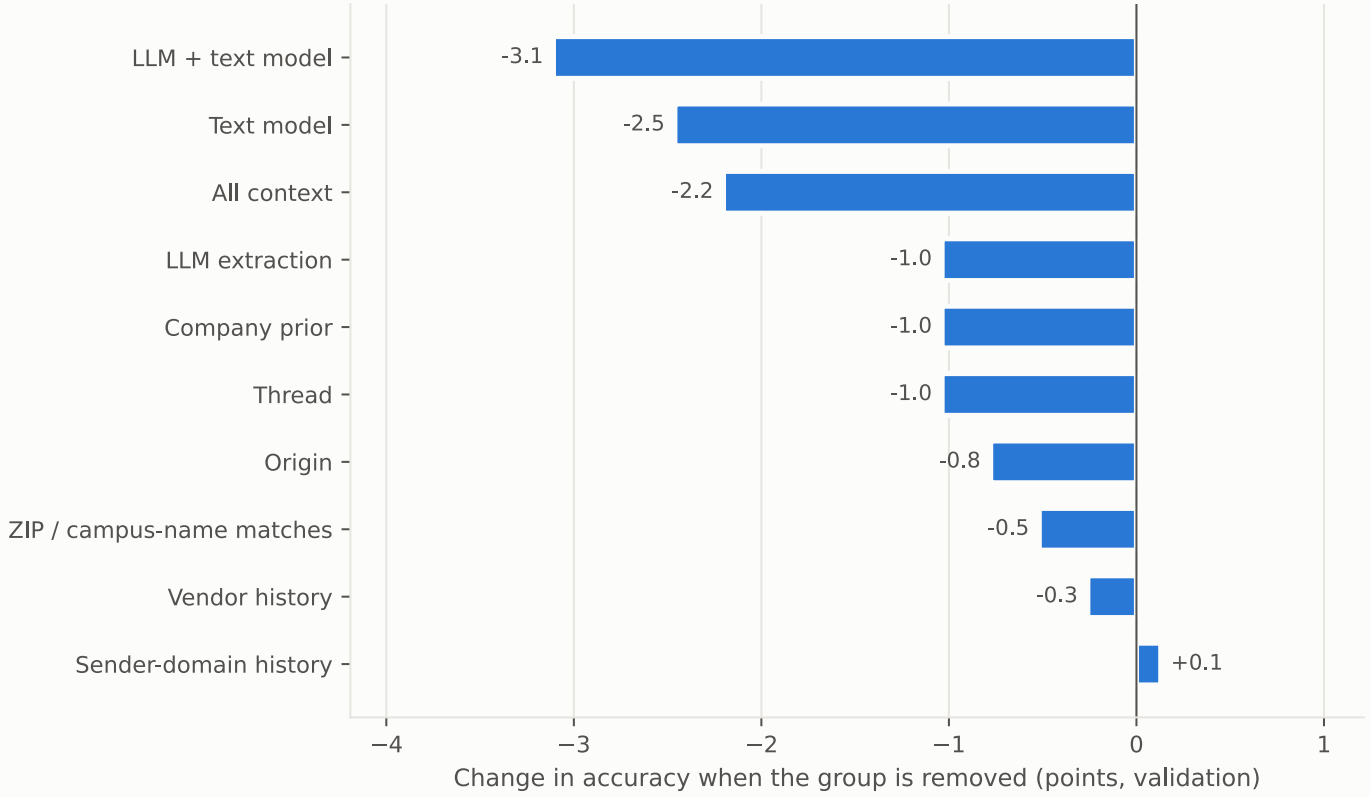
Fit on 4,977 linked messages (January–July), scored on 772 August messages. All design choices were made here.

6. What carries the signal

To see which inputs carry the signal, the candidate ranker was refit on August with one feature group removed at a time. Every single group moves accuracy by only one to three points: the information is spread across overlapping signals (the vendor's name appears in the text, in the LLM's extraction and in the vendor-history features), so no one group is indispensable. Removing the two content models together (LLM and text) costs the

most. The per-slice results are the clearer evidence of what matters: follow-ups and Austin — where history decides — are where the zero-shot methods collapse and the learned ones do not.

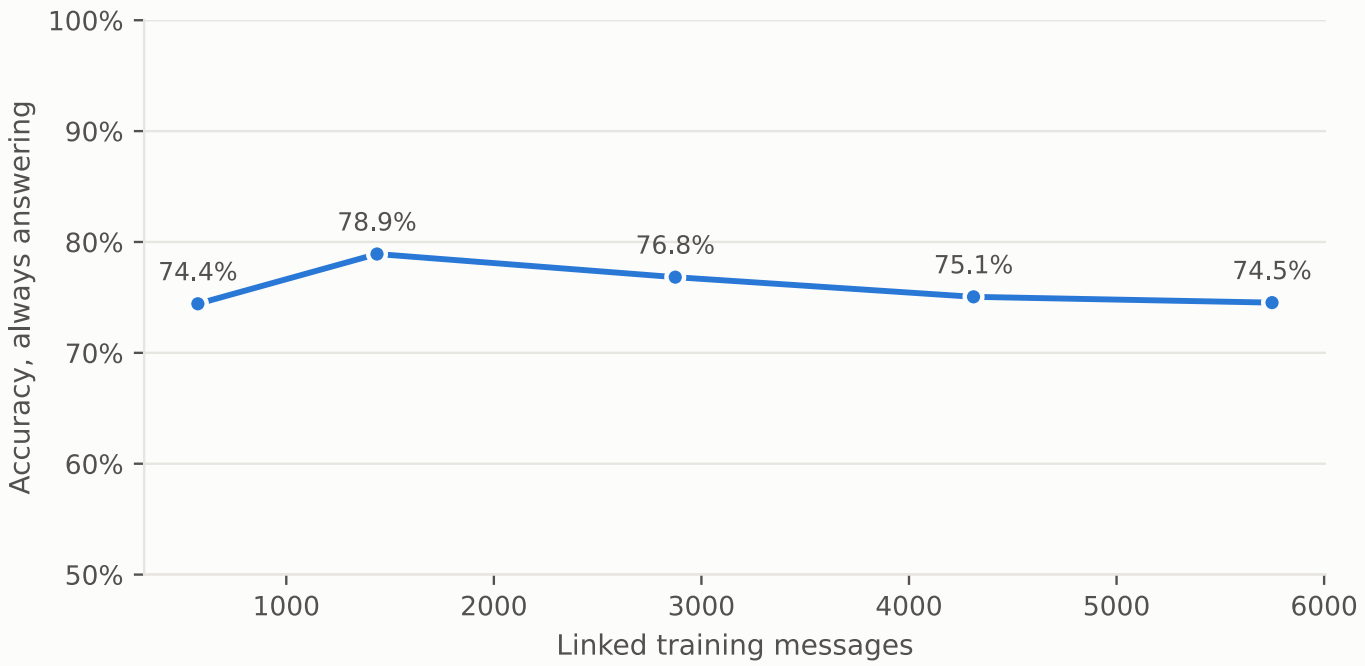
The learning curve (below) was run on the ranker. It does not rise with more data: 10% of the training emails does about as well as all of them. The limit is ambiguity (the HQ address, shared vendors), not volume.



Ablation: the ranker is refit without one feature group at a time and re-scored on August. Full model: 76.8% always answering. Differences under ~1.5 points are within noise.

Feature group removed	Change (points)
LLM + text model	-3.1
Text model	-2.5
All context	-2.2
LLM extraction	-1.0
Company prior	-1.0
Thread	-1.0
Origin	-0.8
ZIP / campus-name matches	-0.5
Vendor history	-0.3
Sender-domain history	+0.1

Does more data help?



Candidate ranker, accuracy always answering on September, trained on a random fraction of the 5,749 January–August labelled messages (fixed seed).

7. Findings

- **Most invoices do not come from the vendor.** 39% of labelled emails are employee forwards, 15% arrive through invoicing platforms (QuickBooks, Stripe, Workday), and only 22% come straight from the vendor. Methods that treat the sender as the vendor are wrong for most of the mailbox.
- **Headquarters is a trap for every method that reads addresses.** 1044 Liberty Park Dr and 1201 Spyglass Dr (Austin 78746, "Legacy of Education, Inc.") appear as the Bill-To on bills booked to Alpha High School and other campuses. The LLM explains its wrong answers with exactly this address.
- **The learned models' blind spot is new and small campuses**, which they absorb into the headquarters campus; Fin District, Marin and Armonk score 0% with gradient boosting alone. Routing to the LLM fixes part of this.
- **The experiment itself needed guarding.** A leaky thread feature, tuning on the test month, and a lenient "any matching company counts" ground truth each flattered earlier results by several points; all three were fixed before the numbers above were produced.
- **Cheap LLM settings are enough.** GLM 5.3 Flash at low reasoning effort matched default effort on the same emails (75.2% vs 75.5% precision, same answer 96% of the time) at one-seventh of the output tokens.

8. Caveats

- **The scored population is emails we could link to a bill** (6,707 of 33,642; 1,151 more were ambiguous and excluded). They are the emails with an invoice number, a matching attachment or an exact amount — probably easier than average. How the methods do on the rest, and on deciding whether an email is an invoice at all, is untested.
- **15 of 73 companies.** Those with the most mail were chosen; the long tail of small campuses — exactly where the learned models are weakest — is under-represented.
- **One test month.** September is 958 emails; confidence intervals are ±2–3 points for the best methods and wider for small slices and companies.
- **The ground truth is built by a matcher** (strongest evidence per email); a wrong vendor+amount match would be a wrong label. 44% of labels rest on an identical attachment and 31% on the invoice number.
- **The HQ-address fix was identified on the test month**, so it has not been applied here; it must be evaluated on validation first.

9. Recommendation and next steps

Recommendation. Build the production classifier as the routing ensemble: gradient boosting over the email, QuickBooks-history and LLM-extracted features, with the LLM allowed to override when it names a specific non-HQ campus at ≥ 0.9 confidence. Auto-assign when the combined confidence is ≥ 0.9 (79% of emails at 91% accuracy in this backtest) and send the rest to a person with the top suggestions. Do not use rules or Jev as the classifier.

Next steps, in order of expected gain:

1. **Teach every component about the headquarters addresses** — as a feature ("Bill-To is an HQ address") and in the LLM prompt — evaluated on August before September is touched again.
2. **Authorize the remaining ~57 companies**, prioritising small and new campuses, so the learned model stops defaulting them to Austin.
3. **Score the non-invoice side**: whether an email is a bill at all, using the AP team's Gmail labels as ground truth.
4. **Account-level prediction** (which expense account), where the chart of accounts is one template across companies.

Appendix: reproducing this

Everything is in the `AlphaSchoolAttgTriage` repository; derived data lives in the project's private S3 bucket (`scripts/data_sync.py pull`). In order:

1. `export_alpha_mail.py --mailbox "[Gmail]/All Mail" --since 2026-01-01 --out mail_archive` — read-only mailbox export

2. `scripts/qbo_discover.py` — every authorized QuickBooks company: accounts, classes, vendors, bills
3. `scripts/qbo_match_mail.py --mail-dir mail_archive` — link emails to bills (the ground truth)
4. `scripts/llm_extract.py --model fw-fireworks-trilogy-glm-5p3-flash` — LLM fields per linked email
5. `scripts/build_backtest.py` — one row per email: as-of features, outcomes, split
6. `scripts/backtest_ml.py`, `scripts/backtest_rank.py` (and `--drop GROUP`, `--fit-frac F`) — learned models, ablations, learning curve
7. `classify_locations_jev.py` (`JEV_CHOICES=active` for the 16-label run) — Jev
8. `scripts/run_experiments.py` — every method scored on the same rows
9. `scripts/build_report.py` — this document

Full method and protocol notes: `docs/approaches.md`; history of experiments: `docs/experiments.md`.