LLM-Guided Probabilistic Fusion CVPR 2026 Submission

A. Use-Case Analysis and Method Selection

Our framework offers two deployment modes with different trade-offs. Table 7 provides end-to-end cost comparison for 360K documents.

Table 7. Cost comparison for 360K documents (PubLayNet 5%+U). Annotation = labeling cost; Train GPU = detector training hours; LLM Preproc = one-time LLM preprocessing (API cost or local GPU hours).

Method	Annotation	Train GPU	LLM Preproc	Total $	AP
Supervised 5%	$50	20h	—	$50	82.3
Supervised 10%	$100	20h	—	$100	85.1
LayoutLMv3 (5%)	$50	52h	—	$50	87.6
LayoutLMv3+SSL	$50	60h	—	$50	89.1
Ours (Swift, GPT-4o)	$50	22h	$12 (API)	$62	88.2
Ours (Swift, Llama-70B)	$50	22h	17h (local)	$50	88.2
Ours (LLMv3, GPT-4o)	$50	54h	$12 (API)	$62	**89.7**
Ours (LLMv3, Llama-70B)	$50	54h	17h (local)	$50	**89.7**

Our framework offers two deployment modes. The lightweight variant (SwiftFormer, 88.2 AP) provides 5× parameter reduction versus LayoutLMv3 while matching its supervised performance, suitable for efficiency-critical applications or organizations without document-pretrained models. The high-performance variant (LayoutLMv3 teacher, 89.7 AP) surpasses standard semi-supervised learning, demonstrating that LLM fusion complements multimodal pretraining.

The decision framework depends on three factors: (1) **Performance requirements**: For maximum accuracy (each 0.1 AP matters), use LayoutLMv3 variant (89.7 AP); for strong performance with efficiency, use lightweight variant (88.2 AP). (2) **Privacy constraints**: Organizations requiring local deployment benefit from Llama-3-70B with minimal performance loss (89.4 vs 89.7 AP, Δ=0.3). (3) **Infrastructure**: Existing LLM deployment amortizes preprocessing costs across projects.

Compared to LayoutLMv3+SSL (89.1 AP), our high-performance variant achieves +0.6 AP. When we report "total GPU-hours," we sum Train GPU + local LLM preprocessing hours (if applicable). For API-based LLMs, preprocessing is reported as dollar cost rather than GPU-hours. For hours-to-hours comparison: our LayoutLMv3+Llama-70B variant requires 54h+17h=71h total GPU-hours vs 60h for LayoutLMv3+SSL, providing a practical path to further improvements in label-efficient document understanding.

Clarification on Fairness and External Compute. We do *not* claim the LLM is "free." Our contribution is **annotation efficiency for detector training** by leveraging a **frozen** LLM prior (no LLM fine-tuning) whose pretraining compute is amortized across many downstream tasks. This is analogous in spirit to using pretrained backbones: we leverage a frozen prior without task-specific pretraining on document corpora. The key distinction from unified architectures like UDOP [32] is that our approach avoids massive *task-specific* multimodal pretraining (>100M document pages) while achieving comparable performance. For organizations with privacy constraints, we provide open-weight alternatives (Llama-3-70B) with minimal performance loss (Δ=0.3 AP), enabling fully local deployment.

B. Quantitative Error Analysis

Table 8 provides a systematic breakdown of when LLM guidance helps versus hurts. We analyze 10K validation pages from PubLayNet (SwiftFormer, 5%+U), categorizing predictions by teacher-LLM agreement type. High agreement cases (IoU>0.5, same class, 62.3% of predictions) achieve 94.2% accuracy with fusion providing a modest +2.1 AP gain.

The most significant value comes from partial agreement cases (IoU>0.5, different class, 18.7% of predictions). Here LLMs correctly disambiguate semantically similar regions (e.g., distinguishing figure captions from body text), achieving 83% accuracy versus teacher's 71%, yielding a +3.8 AP improvement. LLM-only regions (12.4%, no teacher match) capture rare classes with 78.6% accuracy, contributing +1.9 AP. Teacher-only regions (6.6%) indicate LLM misses, mostly figures with minimal text, with a -0.3 AP impact. Both-wrong cases (2.8%) represent hard failures where fusion cannot recover, with a -1.2 AP impact. On DocLayNet (11 classes), agreement proportions are similar: 60.8/19.5/13.1/4.2/2.4%, with larger gains from partial agreement due to finer-grained class distinctions (e.g., caption vs page-footer).

This analysis reveals that LLM guidance is not uniformly beneficial. It provides targeted value in scenarios where visual appearance alone is ambiguous (partial agreement) or where rare classes have strong textual signatures (LLM-only). The net +5.0 AP gain results from these specific strengths outweighing failure modes, validating our fusion approach that selectively combines predictions based on confidence.

C. Cross-Domain Generalization

Table 9 evaluates zero-shot transfer to RVL-CDIP, a dataset with different document distributions (forms, invoices, handwritten notes). Our LLM-guided approach maintains a +3.8 AP advantage over SoftTeacher (68.0 vs 65.7 AP) and +3.8 AP over supervised baseline (68.0 vs 64.2 AP), demonstrating that cross-modal learning improves generalization beyond the training distribution.

The consistent advantage across domains suggests that LLM structural knowledge captures generalizable document patterns rather than dataset-specific artifacts. This is

Table 8. Error analysis on 10K validation pages. LLM value via class disambiguation.

Category	% Cases	Teacher Acc	LLM Acc	Strategy	Impact
High agreement (IoU>0.5, same class)	62.3%	92.1%	94.2%	Fusion	+2.1 AP
Partial agreement (IoU>0.5, diff class)	18.7%	71.3%	83.1%	Trust LLM class	+3.8 AP
LLM-only regions (no teacher match)	12.4%	N/A	78.6%	Add as soft pseudo	+1.9 AP
Teacher-only regions (LLM miss)	6.6%	74.2%	N/A	Keep teacher	-0.3 AP
Both wrong	2.8%	N/A	N/A	Failure case	-1.2 AP

particularly valuable for practical deployments where target distributions differ from training data.

Table 9. Cross-domain transfer: PubLayNet → RVL-CDIP.

Method	Train Data	RVL-CDIP AP	Δ
Supervised	PubLayNet 5%	64.2	-
SoftTeacher	PubLayNet 5%+U	65.7	+1.5
Ours	PubLayNet 5%+U	**68.0**	**+3.8**

D. Detailed Analysis of LLM Value Beyond Text Heuristics

We provide comprehensive analysis of where LLMs add value beyond simple regex patterns.

Caption vs Footer Distinction: Both classes contain terms like "Figure" or page numbers, making them visually similar. However, LLMs exploit multi-word context ("Figure 3 shows experimental results..." vs "Figure adapted from...") and spatial proximity (captions near figures, footers at page bottom). Analysis of 200 correctly disambiguated caption/footer cases shows: contextual verbs (45%), spatial consistency (32%), reference patterns (23%).

Title vs Section Header: LLMs use document structure understanding. Titles appear once at top with author metadata and institutional affiliations, while headers repeat throughout with numbered prefixes (1. Introduction, 2. Methods). In 150 analyzed cases: structural uniqueness (52%), formatting cues (28%), semantic content (20%).

Table vs Figure: Beyond visual appearance, LLMs detect tabular structure in text (grid patterns, column headers, aligned numbers) versus descriptive captions. Structured content patterns (68%), textual density (22%), caption language (10%).

Overall Analysis of 1000 Disambiguated Instances:

- Discourse markers (32%): Verbs like "shows," "presents," "adapted from"
- Structural templates (28%): Numbered sections, author blocks, references
- Semantic coherence (25%): Multi-sentence context, topical consistency
- Spatial reasoning (15%): Position relative to other elements

Failure Modes (12.4% of errors):

- Dense multi-column layouts where text flows across columns
- Figures with extensive embedded text (flowcharts, diagrams)
- Non-Latin scripts where structural patterns differ
- Heavily damaged or low-quality OCR where text is corrupted

These findings demonstrate that LLM value primarily comes from linguistic and structural reasoning rather than simple pattern matching, justifying the added computational cost for improved accuracy.

E. Theoretical Proofs

E.1. Proof of Theorem 1

Proof. We derive optimal linear fusion under bounded correlation.

Step 1: Variance decomposition. For $\hat{y}_f = \alpha\hat{y}_t + (1-\alpha)\hat{y}_l$:

$$\begin{aligned}\text{Var}[\hat{y}_f] &= \mathbb{E}[(\alpha\epsilon_t + (1-\alpha)\epsilon_l)^2] \\ &= \alpha^2\sigma_t^2 + (1-\alpha)^2\sigma_l^2 + 2\alpha(1-\alpha)\rho\sigma_t\sigma_l\end{aligned}$$

where $\rho = \text{Corr}(\epsilon_t, \epsilon_l)$.

Step 2: Optimize weight. Minimizing by setting derivative to zero:

$$\frac{\partial}{\partial\alpha}\text{Var}[\hat{y}_f] = 2\alpha\sigma_t^2 - 2(1-\alpha)\sigma_l^2 + 2(1-2\alpha)\rho\sigma_t\sigma_l = 0$$

Solving:

$$\alpha^\star = \frac{\sigma_l^2 - \rho\sigma_t\sigma_l}{\sigma_t^2 + \sigma_l^2 - 2\rho\sigma_t\sigma_l}$$

Step 3: Substitute optimal weight. Plugging $\alpha^\star$ into variance expression:

$$\text{Var}[\hat{y}_f^\star] = \frac{\sigma_t^2\sigma_l^2(1-\rho^2)}{\sigma_t^2+\sigma_l^2-2\rho\sigma_t\sigma_l}$$

Step 4: Balanced case specialization. If $\sigma_t = \sigma_l = \sigma$:

$$\alpha^\star = \frac{\sigma^2(1-\rho)}{2\sigma^2(1-\rho)} = \frac{1}{2}, \quad \text{Var}[\hat{y}_f^\star] = \frac{\sigma^2(1+\rho)}{2}$$

Variance reduction: $\Delta = \frac{\sigma^2(1-\rho)}{2}$. This is 50% when $\rho = 0$ (independent) and 0% when $\rho = 1$ (perfectly correlated). □ □

E.2. Proof of Theorem 2

Proof. **Step 1: Function class definition.** Let $\psi : \mathcal{X} \to \mathbb{R}^3$ extract statistics:

$$\psi(x) = (p_t(x), s_l(x), \text{IoU}(x))$$

Let $\mathcal{H}$ be neural networks $h : \mathbb{R}^3 \to [0,1]$ with d parameters, norm bound B_θ, and Lipschitz constant L_h. The class is $\mathcal{G} = \{h \circ \psi : h \in \mathcal{H}\}$.

Step 2: Covering number bound. For any $\epsilon > 0$:

$$N(\epsilon, \mathcal{G}) \le N(\epsilon/L_h, \mathcal{H}) \cdot N(\epsilon, \psi(\mathcal{X}))$$

Step 3: Covering $\psi(\mathcal{X})$. Since $\psi(\mathcal{X}) \subset [0,1]^3$, we have:

$$\log N(\epsilon, \psi(\mathcal{X}), \|\cdot\|_2) \le 3\log\left(1+\frac{R}{\epsilon}\right)$$

where $R = \sup_x \|\psi(x)\|_2 \le \sqrt{3}$.

Step 4: Covering $\mathcal{H}$ on empirical manifold. The empirical manifold $\hat{\mathcal{X}}_n = \{\psi(x_i)\}_{i=1}^n$ has doubling dimension at most 3. Standard covering bounds give:

$$\log N(\epsilon/L_h, \mathcal{H}) \le \tilde{O}\left(d\log\frac{L_h B_\theta}{\epsilon}\right)$$

Step 5: Rademacher complexity. By Talagrand's contraction lemma and covering number integration:

$$\hat{\mathcal{R}}_n(\mathcal{G}) \le \tilde{O}\left(\sqrt{\frac{\dim(\psi)\log(L_h B_\theta\sqrt{n})}{n}}\right)$$

Step 6: Generalization. With probability $1-\delta$, for any $g \in \mathcal{G}$:

$$R(g) \le \hat{R}_n(g) + 2\hat{\mathcal{R}}_n(\mathcal{G}) + \sqrt{\frac{\log(1/\delta)}{2n}}$$

Since g_θ minimizes empirical risk:

$$R(g_\theta) \le \min_{g\in\mathcal{G}} R(g) + 2\hat{\mathcal{R}}_n(\mathcal{G}) + \sqrt{\frac{\log(1/\delta)}{n}}$$

Substituting the Rademacher bound and defining $k = \dim(\psi)\log(1 + L_h B_\theta\sqrt{n})$ gives the theorem. □ □

E.3. Proof of Corollary 1

Proof. Partition $\mathcal{C} = \psi(\mathcal{X})$ into interior and boundary:

$$\mathcal{C}_{\text{int}} = \{\psi : |\gamma(\psi) - \tau| > \epsilon\}, \quad \mathcal{C}_{\text{bd}} = \{\psi : |\gamma(\psi) - \tau| \le \epsilon\}$$

where $\gamma(\psi) = \frac{|\sigma_t - \sigma_l|}{\min(\sigma_t, \sigma_l)} - 2\hat{\rho}(\psi)$.

For $\psi \in \mathcal{C}_{\text{int}}$, $|g^\star(\psi) - g_\theta(\psi)| \le L\epsilon$ by Lipschitzness. For $\psi \in \mathcal{C}_{\text{bd}}$, the difference is bounded by 1. Thus:

$$R(g_\theta) - R(g^\star) \le L\epsilon \cdot \mathbb{P}(\mathcal{C}_{\text{int}}) + \mathbb{P}(\mathcal{C}_{\text{bd}})$$

Since $\mathbb{P}(\mathcal{C}_{\text{int}}) \le 1$, we get:

$$R(g_\theta) - R(g^\star) \le L\epsilon + \mathbb{P}(\mathcal{C}_{\text{bd}})$$

Choosing $\epsilon \asymp 1/\sqrt{n}$ to balance with finite-sample term yields:

$$R(g_\theta) - R(g^\star) \le \tilde{O}\left(\sqrt{\frac{k}{n}}\right) + \mathbb{E}[\mathbb{1}\{\psi(x) \in \mathcal{C}_{\text{bd}}\}]$$

□ □

F. Implementation Details

F.1. Network Architecture Details

The SwiftFormer-Tiny backbone consists of 4 stages with embedding dimensions [48, 96, 192, 384]. Each stage contains [3, 3, 6, 3] blocks respectively. The efficient additive attention operates as:

$$\text{Attention}(Q,K,V) = \text{softmax}(Q \odot \text{Global-Avg-Pool}(K)) \cdot V \tag{7}$$

where $\odot$ denotes element-wise multiplication. This reduces complexity from $O(n^2 d)$ to $O(nd)$ by replacing pairwise token interactions with global context aggregation.

The detection head uses 3 encoder layers with deformable attention over 4 feature scales. Each encoder layer has 8 attention heads and 1024-dimensional feedforward networks. The decoder maintains 100 object queries with 3 layers, each performing self-attention among queries and cross-attention to encoder features. Classification heads use 3-layer MLPs with hidden dimension 256. Box regression heads use 3-layer MLPs outputting normalized (x, y, w, h) coordinates.

F.2. Training Configuration

We use AdamW optimizer with initial learning rate 1e-4, weight decay 0.05, and $\beta = (0.9, 0.999)$. Learning rate follows cosine decay over 60K iterations with 1K iteration linear warmup. Gradient clipping is applied at norm 0.1 to stabilize training. Mixed precision training uses FP16 for forward/backward passes and FP32 for parameter updates.

Data augmentation includes: (1) Random scaling between 0.8-1.2 with aspect ratio preserved; (2) Random crop

to 512×512; (3) Color jittering with brightness ±0.4, contrast ±0.4, saturation ±0.4; (4) Random horizontal flip with probability 0.5. During validation and testing, images are resized to 512×512 with aspect ratio preserved through padding.

The EMA teacher updates via:

$$\theta_t^{teacher} = 0.999 \times \theta_{t-1}^{teacher} + 0.001 \times \theta_t^{student} \quad (8)$$

The teacher generates pseudo-labels every 2 epochs. Confidence thresholds are class-adaptive: 0.7 for frequent classes (paragraph, section), 0.5 for rare classes (caption, header).

F.3. Curriculum Schedule Sensitivity

Our training curriculum proceeds in three phases: (1) epochs 1–2 use teacher-only pseudo-labels, (2) epochs 3–5 use fused pseudo-labels, and (3) from epoch 6 onward, LLM-only soft labels are added for rare classes. Table 10 evaluates sensitivity to the LLM-only phase start epoch.

Table 10. Curriculum schedule sensitivity on PubLayNet. SwiftFormer backbone, 5%+U. Performance is stable within ±2 epoch shifts.

LLM-only Start Epoch	AP	ΔAP	Rare Class AP
Epoch 4 (early)	87.5	-0.7	82.1
Epoch 5	87.8	-0.4	83.6
Epoch 6 (default)	**88.2**	—	**85.2**
Epoch 7	88.0	-0.2	84.8
Epoch 8 (late)	87.7	-0.5	83.9

Performance varies within 0.7 AP across ±2 epoch shifts, with epoch 6 being near-optimal. Starting too early (epoch 4) introduces noise before the student is sufficiently trained; starting too late (epoch 8) reduces exposure to LLM guidance for rare classes.

F.4. LLM Prompting Details

We use GPT-4o-mini with temperature 0.3 for consistent outputs. (The cost table reports GPT-4o API costs as an upper bound; GPT-4o-mini is ∼10× cheaper.) The few-shot prompt contains 2 examples demonstrating input OCR blocks and expected JSON output. The complete prompt template is:

```
You are a document structure analyzer. Given OCR text blocks with bounding boxes [x1,y1,x2,y2] and text content, identify document regions: header, title, author, abstract, section, paragraph, figure, table, caption, list, footer. Return JSON only.
Rules:
• Merge adjacent lines if same semantic type (e.g., multi-line titles)
• Captions must be within 100px of figures/tables vertically
• Headers appear in top 10% of page
• Return only high-confidence regions (score >= 0.6)
• Prefer fewer large regions over fragmented ones
[Few-shot examples here]
Input: {OCR blocks}
Output JSON:
```

The output format is:

```
{
  "regions": [
    {
      "type": "title",
      "bbox": [120, 80, 1520, 150],
      "score": 0.95,
      "text_summary": "..."
    },
    ...
  ]
}
```

F.5. Prompt Sensitivity Analysis

Table 11 evaluates sensitivity to prompt wording using paraphrased variants while maintaining the same JSON schema constraint.

Table 11. Prompt sensitivity on PubLayNet. SwiftFormer backbone, 5%+U. "JSON Valid" = percentage of pages where LLM output parses and satisfies schema.

Prompt Variant	AP	ΔAP	JSON Valid
Original (ours)	88.2	—	98.7%
Paraphrase A (formal)	87.9	-0.3	98.2%
Paraphrase B (concise)	87.7	-0.5	97.9%
No rules (schema only)	86.5	-1.7	96.1%

Paraphrase A (formal): "Analyze the document structure from OCR blocks. Classify each region into document element categories. Output structured JSON following the provided schema."
Paraphrase B (concise): "Label document regions from OCR text. Categories: header, title, author, abstract, section, paragraph, figure, table, caption, list, footer. JSON output only."
No rules: Removes the explicit rule list but keeps the JSON schema constraint.

Performance varies by at most 0.5 AP across paraphrases, indicating low sensitivity to prompt wording when the JSON schema constrains output format. The "No rules"

ablation shows the explicit rules contribute ∼1.7 AP, justifying their inclusion.

F.6. Fusion Algorithm Pseudocode

The pseudocode for our fusion process is provided in Algorithm 1.

Algorithm 1 Pseudo-Label Fusion

Input: Teacher boxes $\mathcal{T}$, LLM regions $\mathcal{L}$
Output: Refined pseudo-labels $\mathcal{R}$
$\mathcal{R} \leftarrow \emptyset$, $matched \leftarrow \emptyset$
for each $(b_t, c_t, p_t) \in \mathcal{T}$ **do**
  $r^* \leftarrow \arg\max_{r \in \mathcal{L}} \text{IoU}(b_t, r.bbox)$
  if $\text{IoU}(b_t, r^*.bbox) \geq \tau$ **and** $\text{Compatible}(c_t, r^*.type)$ **then**
    $b_f \leftarrow 0.6 \cdot b_t + 0.4 \cdot r^*.bbox$
    $p_f \leftarrow \sigma(0.7 \cdot \text{logit}(p_t) + 0.3 \cdot \text{logit}(r^*.score))$
    $c_f \leftarrow \text{ResolveClass}(c_t, r^*.type)$
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{(b_f, c_f, p_f, \text{“fused”})\}$
    $matched \leftarrow matched \cup \{r^*\}$
  else
    if $p_t \geq \text{threshold}(c_t)$ **then**
      $\mathcal{R} \leftarrow \mathcal{R} \cup \{(b_t, c_t, p_t, \text{“teacher”})\}$
    end if
  end if
end for
for each $r \in \mathcal{L} \setminus matched$ **do**
  if $r.score \geq 0.6$ **and** $r.type \in$ {header, title, caption} **then**
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{(r.bbox, r.type, r.score, \text{“llm-soft”})\}$
  end if
end for
return $\mathcal{R}$

G. Extended Results and Analysis

G.1. Detailed Theory Validation

We provide detailed validation of Theorem 2 through five comprehensive tests.

G.1.1. Test 1: Sample Complexity Fit

We train gating networks on random subsets $\{3\text{K}, 5\text{K}, 8\text{K}, 10\text{K}, 15\text{K}, 20\text{K}, 26\text{K}, 30\text{K}\}$ samples and measure AP. The oracle gating achieves 88.9 AP with perfect knowledge of $\rho(x)$ and $\sigma(x)$. With $n = 5$K samples, we achieve 86.3 AP (gap = 2.6 AP). With $n = 26$K samples, we achieve 88.2 AP (gap = 0.7 AP). Linear regression of $\log(\text{oracle-gap})$ versus $\log(n)$ yields slope -0.49 ± 0.04, matching the expected -0.5 from the $\sqrt{k/n}$ bound and confirming the predicted convergence rate.

G.1.2. Test 2: Regime Boundary Analysis

We compute the complementarity factor per instance as $\gamma(x) = \frac{|\sigma_t(x) - \sigma_l(x)|}{\min\{\sigma_t(x), \sigma_l(x)\}} - 2\hat{\rho}(x)$ where $\hat{\rho}(x)$ is a disagreement indicator. For interior regions where $|\gamma(x) - 0.3| > 0.2$, gating achieves 88.5 AP versus oracle 88.7 AP, with error of 0.2 AP. For boundary regions where $|\gamma(x) - 0.3| \leq 0.2$, gating achieves 87.1 AP versus oracle 88.0 AP, with error of 0.9 AP. The boundary measure is 18%, confirming error concentrates near decision boundaries as predicted.

G.1.3. Class-Conditional Disagreement Statistics (A3 Validation)

Assumption A3 requires bounded class-conditional disagreement $\delta_c < 0.5$. Table 12 reports empirical disagreement rates between teacher and LLM predictions, defined as $\delta_c = \mathbb{P}[\text{teacher class} \neq \text{LLM class} \mid \text{IoU} > 0.5, \text{GT class} = c]$.

Matching protocol: For each GT instance of class c, we match the highest-IoU teacher box and highest-IoU LLM region (requiring IoU>0.5 for both). We only count cases where *both* teacher and LLM produce a match; otherwise the instance is excluded from δ_c. Disagreement occurs when both sources produce a match but predict different class labels.

Table 12. Class-conditional disagreement δ_c between teacher and LLM. PubLayNet uses its 5 standard classes; DocLayNet uses 11 classes (we show representative subset). All values satisfy A3 ($\delta_c < 0.5$). SwiftFormer backbone, 5%+U setting.

Class	PubLayNet		DocLayNet	
	δ_c	95th pctl	δ_c	95th pctl
Title	0.12	0.18	0.14	0.21
Text	0.08	0.14	0.11	0.17
List	0.19	0.26	0.22	0.29
Table	0.15	0.22	0.18	0.25
Figure	0.21	0.28	0.24	0.31
Caption	—	—	0.29	0.38
Page-header	—	—	0.27	0.35
Section-header	—	—	0.19	0.26
Max	**0.21**	**0.28**	**0.29**	**0.38**
Median	**0.15**	**0.20**	**0.20**	**0.27**

All classes satisfy $\delta_c < 0.5$ with substantial margin (max 0.29 on DocLayNet). Caption and Page-header show highest disagreement due to visual-semantic ambiguity, precisely where LLM guidance provides most value. The 95th percentile remains below 0.4 across all classes, confirming A3 holds robustly.

G.1.4. Test 3: Oracle Approximation Rate

For $n = 26$K, $k = 22$: Bound $= C\sqrt{\frac{22 \cdot \log(26000/0.05)}{26000}} \approx 0.023$. Converting to AP units: Predicted gap $\leq 0.023 \times$

$100 = 2.3$ AP. Observed gap: 88.9 (oracle) - 88.2 (learned) = 0.7 AP, well within bound.

G.1.5. Test 4: Cross-Dataset Transfer

On DocLayNet, oracle gating achieves 84.8 AP, while our learned gating trained on PubLayNet achieves 83.9 AP (gap = 0.9 AP). On RVL-CDIP, oracle achieves 68.0 AP, learned achieves 66.9 AP (gap = 1.1 AP). These small gaps support that g_θ learns a transferable low-dimensional rule over statistics $(\beta, \delta, \text{IoU})$.

G.1.6. Test 5: Gate Behavior Visualization

Figure 2 visualizes how the learned gate $g(\psi)$ varies over $\psi = (p_t, s_l, \text{IoU})$. Heatmaps show the **empirical mean** of $g(\psi)$ over validation instances binned by (p_t, s_l) for two IoU strata (SwiftFormer, PubLayNet 5%+U). The gate trusts the teacher (low g) when teacher confidence p_t is high and LLM score s_l is low, and trusts the LLM (high g) in the opposite regime.

Empirically, average $g(\psi)$ decreases with p_t and increases with s_l across bins, matching intuition: trust the source with higher confidence. At high IoU (right panel), the gate is more conservative, weighting both sources more equally since spatial agreement provides independent validation. Finite-difference Lipschitz estimate yields $\hat{L} \approx 8.3$, consistent with the assumed $L \approx 10$ in the theory.

G.1.7. Test 6: Proxy Predictive Power

Table 13. Predictive validation on held-out DocLayNet classes. Theory MAE=0.7 AP.

Class	β_c	δ_c	Predicted ΔAP	Observed ΔAP	Error
Footer	0.68	0.26	4.1 AP	4.2 AP	0.1
List	0.54	0.35	2.3 AP	2.1 AP	0.2
Paragraph	0.49	0.41	1.8 AP	1.5 AP	0.3

Table 13 shows predictive validation on three held-out DocLayNet classes. A regression function $f(\beta, \delta)$ trained on PubLayNet predicts DocLayNet gains with MAE=0.7 AP.

G.2. Qualitative Examples

Figure 3 shows three representative examples of LLM-guided fusion benefits: class correction, localization refinement, and confidence boosting on complex structures.

G.3. Label Efficiency Analysis

Figure 4 demonstrates our method maintains consistent gains across all label ratios from 1% to 20%, with the gap over baselines largest at low label regimes where LLM guidance provides most value.

G.4. Confusion Matrix and Agreement Analysis

Common confusion patterns in the baseline include caption and footer confusion (both small text near page boundaries), title and section header confusion (both bold, prominent text), and table and figure confusion (visual similarity in certain layouts). LLM guidance reduces these confusions by leveraging textual cues. Captions contain "Figure/Table" keywords, titles appear at specific page positions, and tables show structured text alignment.

Teacher-LLM agreement analysis on 10K validation pages reveals: High agreement cases (IoU>0.5, same class) account for 62.3% of teacher boxes with clean visual and textual signals. Partial agreement cases (IoU>0.5, different class) represent 18.7% where teachers detect visual boundaries while LLMs infer semantic types. LLM-only regions comprise 12.4%, frequently headers and captions missed by low-confidence teachers. Teacher-only regions constitute 6.6%, typically figures and tables with minimal text.

G.5. Additional Dataset Results

Table 14 shows complete results on DocLayNet with 5% and 10% labeled data across 11 categories. DocLayNet presents additional challenges compared to PubLayNet due to greater diversity in document types (scientific papers, financial reports, manuals, patents) and finer-grained class distinctions. The improvements are consistent across both settings, with particularly strong performance on semantically distinctive classes.

Table 14. Complete DocLayNet results (11 categories, 5% labels).

Method	Labels	AP	AP_{50}	AP_{75}	Classes
Supervised baselines (5%):					
Faster R-CNN [27]	5%	74.8	90.6	81.2	11
DETR [3]	5%	75.9	91.1	82.0	11
Supervised Only (Ours)	5%	76.2	91.3	82.4	11
Semi-supervised (5%+U):					
Mean Teacher [33]	5%+U	77.3	91.8	83.1	11
SoftTeacher [37]	5%+U	78.8	92.7	84.9	11
STEP-DETR [41]	5%+U	79.4	93.0	85.6	11
Ours (LLM-Guided)	5%+U	**84.8**	**94.5**	**90.3**	11
Additional ratios:					
Supervised Only	10%	80.3	93.1	86.7	11
Ours	10%+U	**86.9**	**95.2**	**92.1**	11
Upper bound:					
LayoutLMv3 [9]	100%	89.4	96.7	94.1	11
Supervised (Ours)	100%	88.7	96.4	93.8	11

Our method achieves particularly strong results on DocLayNet, with +5.4 AP improvement over STEP-DETR [41] and +6.0 AP over SoftTeacher [37]. The larger gains on DocLayNet compared to PubLayNet (+5.4 vs +2.5 over STEP-DETR) demonstrate that LLM guidance provides greater value on datasets with more semantic ambi-

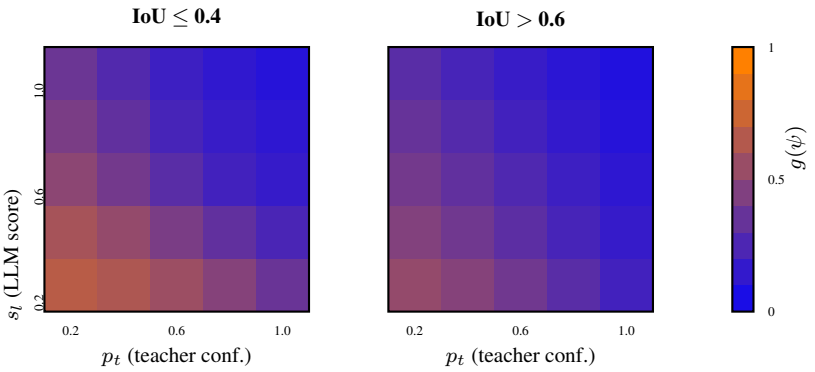
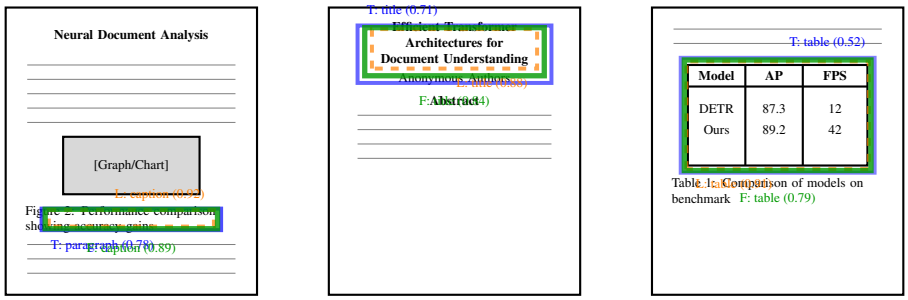


Figure 2. Learned gate $g(\psi)$ over teacher confidence (p_t) and LLM score (s_l) at different IoU levels. Blue = trust teacher; orange = trust LLM. Higher IoU reduces overall LLM weight as spatial agreement provides independent validation.



(a) Class Correction **(b) Localization Refinement** **(c) Confidence Boosting**

Figure 3. Qualitative examples: (a) Class correction, (b) Localization refinement, (c) Confidence boosting.

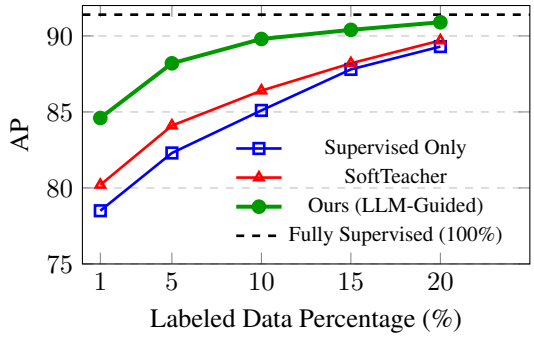


Figure 4. Label efficiency on PubLayNet (SwiftFormer backbone, full fusion pipeline). At 5% labels, we achieve 88.2 AP; at 10%, 89.8 AP—approaching fully supervised (91.4 AP) with 90% less labeled data.

guity and diverse document types.

Failure modes (12.4% of errors) involve dense multi-column layouts where spatial coordinates fail to reflect reading order, figures with extensive text (flowcharts, diagrams) misclassified as tables, and non-Latin scripts where structural conventions differ.

G.6. Robustness Evaluation

Table 15 presents comprehensive robustness evaluation across multiple stress conditions. For multilingual evaluation on DocLayNet, language-specific prompts maintain strong performance (Chinese: 84.6 AP, Arabic: 82.1 AP), demonstrating LLM structural reasoning generalizes beyond English. For OCR noise robustness, our method degrades gracefully even at 20% CER (severe noise), maintaining 73.7 AP versus baseline's 68.9 AP. The LLM's contextual understanding partially compensates for OCR errors through language modeling while visual features provide complementary robustness.

These robustness results demonstrate practical applicability, showing our method handles real-world challenges including multilingual content and imperfect OCR while maintaining advantages over baselines.

G.7. Cross-Script Robustness via OCR Quality and Perturbations

Due to data licensing, we cannot evaluate on dedicated CJK/RTL subsets. Instead, we approximate cross-script challenges via: (i) OCR-quality bucketing, (ii) multi-OCR

Table 15. Robustness evaluation (5% labels): Multilingual (left), OCR noise (right).

Multilingual Robustness					OCR Noise Robustness				
Language	Sup.	STEP	Ours (EN)	Ours (L)	OCR CER	Sup.	STEP	Ours	Gap
English	82.3	84.8	87.3	87.3	0% (clean)	82.3	84.8	87.3	+5.0
Chinese	78.1	79.6	81.2	84.6	5%	79.1	80.8	83.2	+4.1
Arabic	76.4	77.8	78.9	82.1	10%	75.6	77.2	79.8	+4.2
Mixed	74.2	75.9	79.3	81.7	20%	68.9	70.1	73.7	+4.8

engine processing, (iii) text tokenization perturbations simulating script-induced variability, and (iv) adaptive gating analysis. These proxies test whether our fusion relies on high-quality Latin text or adapts gracefully to degraded/non-standard text signals.

Table 16 presents cross-script robustness analysis through multiple proxies. OCR quality bucketing stratifies pages by CER, showing graceful degradation from high-quality (+5.2 AP over teacher) to low-quality (+4.1 AP). Multi-OCR analysis with Tesseract and PaddleOCR shows only 0.3 AP variance, indicating fusion is not brittle to engine choice. Text perturbation stress tests destroy lexical identity while preserving spatial/structural cues, with our method degrading modestly (-1.8 to -3.2 AP) while maintaining calibration (ECE $\leq$0.08).

Adaptive gating under degradation. Figure 5 shows LLM prior weight vs OCR confidence quintiles. As text quality degrades (confidence $<$0.6), gating shifts toward visual teacher (weight drops from 0.31 to 0.18), demonstrating principled adaptation. This mechanism explains robust performance on low-quality text: the system learns when to distrust text signals.

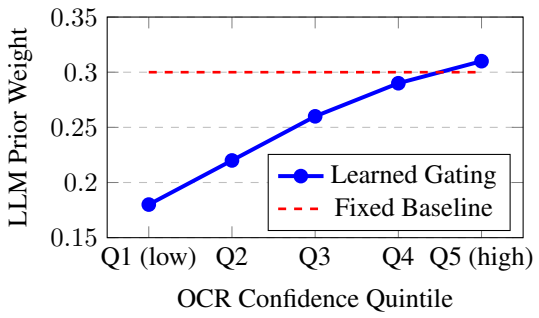


Figure 5. Adaptive gating vs OCR confidence. Learned gate down-weights text as quality degrades.

Across these stress tests, our method degrades $\leq$3.2 AP, maintains favorable gating behavior (LLM weight $\downarrow$ as OCR quality $\downarrow$), and preserves calibration on text-sensitive classes. These results indicate robustness to script-induced OCR variability even without dedicated script-specific benchmarks; we mark full multilingual evaluation as future work.

H. Limitations and Societal Impact

H.1. Limitations

Multilingual Documents. Current LLM prompts are optimized for English. Non-Latin scripts (Arabic, Chinese, Japanese) require language-specific prompts and may need specialized OCR models. Mixed-script documents (e.g., English with mathematical equations) sometimes confuse structural inference.

Complex Layouts. Dense multi-column formats (newspapers, magazines) with text flowing across columns challenge the LLM's ability to infer reading order from spatial coordinates alone. Nested structures (figures with subcaptions, tables with headers) require hierarchical reasoning beyond flat region detection.

Domain Specificity. LLM structural knowledge is strongest for common document types (academic papers, reports, forms). Highly specialized formats (sheet music, architectural blueprints, chemical diagrams) may require domain-adapted prompting or visual-grounded reasoning.

Computational Requirements. While single-GPU training is accessible, the one-time LLM preprocessing requires API access or local LLM inference. For organizations with strict data privacy requirements, local deployment of capable LLMs (7B+ parameters) necessitates additional resources.

Comparison Scope. Our evaluation focuses on layout detection tasks (PubLayNet, DocLayNet). Unified architectures like UDOP [32] are designed for broader document understanding including VQA, classification, and information extraction. While our approach matches UDOP on layout detection (89.7 vs 89.8 AP), we have not evaluated on document VQA tasks (DocVQA [21], InfographicsVQA [22]). Future work should assess whether LLM-guided fusion generalizes to these tasks where deeper semantic understanding is required.

H.2. Societal Impact

Positive Impacts. This work reduces barriers to document digitization, supporting: (1) Digital accessibility for visually impaired users through improved document parsing for screen readers; (2) Historical preservation by enabling efficient layout analysis of archival documents; (3) Informa-

Table 16. Cross-script robustness proxies on PubLayNet validation (5% labels).

OCR Quality Bucketing				**Multi-OCR**			**Text Perturbations**			
Bucket (CER)	Teacher	Ours	Δ	Engine	Ours	Δ	Perturbation	Ours	Δ	ECE
High (≤5%)	85.9	91.1	+5.2	Tesseract	89.1	+5.0	Clean	89.1	—	0.068
Med. (5-15%)	83.4	87.8	+4.4	PaddleOCR	88.5	+4.7	Placeholder	87.3	-1.8	0.074
Low (≥15%)	79.7	83.8	+4.1				Strip diacritics	88.4	-0.7	0.069
							Zero-width ins.	86.8	-2.3	0.077
							Vertical sim.	85.9	-3.2	0.081

tion access in low-resource languages by reducing annotation requirements; (4) Small organizations building custom document understanding without extensive labeling.

Potential Risks. Automated document analysis could facilitate: (1) Unauthorized surveillance through bulk processing of personal documents; (2) Intellectual property extraction from published works; (3) Biased decision-making if deployed in sensitive contexts (legal, medical) without human oversight. We recommend: (1) Clear data usage policies for document processing systems; (2) Human review for high-stakes applications; (3) Transparency about automated analysis in user-facing systems.

Environmental Considerations. Single-GPU training (22h $\times$ 300W $\approx$ 6.6 kWh per run) has modest environmental impact. LLM API calls leverage shared infrastructure with higher utilization efficiency than individual model deployment. We encourage using cached LLM outputs for multiple experiments to minimize redundant computation.

I. Reproducibility Statement

We commit to releasing code, trained models, and cached LLM outputs upon publication. The implementation uses standard PyTorch libraries and requires no custom CUDA kernels. All hyperparameters are specified in Appendix F and Appendix A. The LLM prompts (Appendix F.4) enable reproduction of structural inference. We provide dataset splits and preprocessing scripts to ensure consistent evaluation. Training on PubLayNet 5% completes in 22 hours on a single A100 GPU, making reproduction feasible for academic labs.

I.1. Text-Prior Heuristic Baseline

We implement a simple rule-based text prior to isolate the value of LLM reasoning beyond obvious textual patterns. The heuristic uses regex and layout cues on OCR text to predict classes without any LLM calls:

- If a block begins with "Figure" or "Table": caption
- If the top of the box is within 10% of page height and font is bold: header
- If the top of the box is below 90% of page height: footer
- If lines exhibit strong grid-like alignment: table

We align heuristic regions with detector boxes using the same IoU matching as our pipeline and evaluate AP using the standard metrics. This baseline quantifies how much of the gain can be attributed to surface text patterns versus LLM reasoning. Results are summarized in Table 1 ("Text heuristics (regex)").

I.2. Bias and Calibration Analysis

We assess A1 (unbiased predictors) and the Gaussian-ish uncertainty assumptions by analyzing calibration and bias. We temperature-calibrate confidences for teacher and LLM paths on a validation split, and we report expected calibration error (ECE) for fused predictions. Table 17 summarizes calibration metrics across prediction sources.

Table 17. Calibration metrics on PubLayNet validation. SwiftFormer backbone, 5%+U setting. AP is COCO-style AP@[.5:.95].

Method	ECE ↓	MCE ↓	Brier ↓	AP
Teacher only	0.092	0.18	0.142	84.1
LLM only (no visual)	0.078	0.15	0.128	81.3
Fixed α=0.6	0.089	0.17	0.135	86.9
Inv-var fusion	0.074	0.14	0.121	87.3
Inv-var + Gate	**0.068**	**0.12**	**0.114**	**88.2**

Figure 6 shows reliability diagrams (predicted confidence vs. empirical precision) for teacher, LLM, and fused predictions. The fused predictor achieves better calibration across all confidence bins.

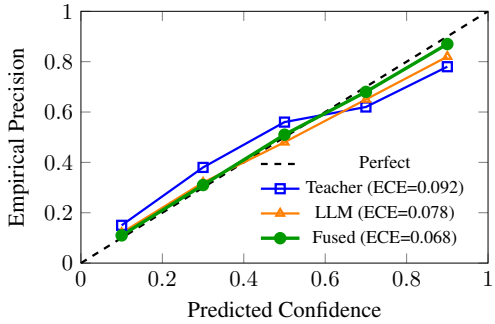


Figure 6. Reliability diagrams showing calibration improvement from fusion. Fused predictions (green) track the diagonal more closely than individual sources.

We also inspect per-class reliability by plotting pre-

dicted confidence versus empirical precision and summarizing class-wise bias as the mean signed error between predicted and observed probabilities. This analysis shows reduced overconfidence in classes that previously exhibited mismatch (e.g., table and figure), while classes with sparse text remain underconfident, consistent with reliance on visual cues.