Beyond Missing Rates: Rethinking Incomplete Multi-View Clustering with Protocol Divergence

Haolu Liu[1] **Xiyue Wang**[1] **Xuanting Xie**[1]
Liangjian Wen[2] **Zhao Kang**[1*]
[1]School of Computer Science and Engineering,
University of Electronic Science and Technology of China
[2]School of Computing and Artificial Intelligence,
Southwestern University of Finance and Economics
{2023080136006,2024080902018}@std.uestc.edu.cn
x624361380@outlook.com wenlj@swufe.edu.cn zkang@uestc.edu.cn

Abstract

Incomplete multi-view clustering (IMVC) is typically evaluated by retraining separate models under different missing-view configurations. Evaluations indexed only by nominal missing rate can overlook differences in observation structure across missing-view protocols. We show that missing-data protocols with identical nominal missing rates can induce substantially different learning regimes, differing by approximately $50\times$ in the proportion of fully observed samples. We formalize this phenomenon as **protocol divergence**, which quantifies structural disparities among missing-view protocols beyond marginal missing rates. Furthermore, we analyze support-gated reconstruction mechanisms and show that their optimization contribution is inherently limited by the frequency of eligible observations under explicit normalization and optimization conditions. Based on these observations, we propose **CRAFT** (Co-occurrence-free Robust Attention-masked Fusion Transformer), a train-once framework that combines representation learning with an architecture designed to process missing-view inputs. CRAFT combines (i) per-sample forward computation using each sample's observed views and shared parameters, and (ii) mask-aware fusion over nonempty observed-view subsets. The deployment evaluation starts from training data with all views available and reuses one final checkpoint per dataset and seed across missing-view protocols without retraining. Experiments on CUB and MultiFashion show that CRAFT achieves the strongest performance in 12 out of 13 information-matched settings. Additional deployment experiments across seven benchmarks and sixteen missing configurations demonstrate substantial computational savings through checkpoint reuse while maintaining competitive clustering performance. Code and evaluation tools: github.com/dk23lhl/CRAFT and github.com/dk23lhl/imvc-audit.

1 Introduction

Multi-view representations have become increasingly important for learning from heterogeneous data collected by multiple sensors, modalities, or feature sources. By exploiting complementary information across views, multi-view clustering (MVC) has demonstrated strong potential for unsupervised representation learning and data discovery. However, the complete-view assumption underlying most MVC methods rarely holds in real-world applications. Sensor failures, privacy

*Corresponding author: Zhao Kang.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

constraints, and transmission errors frequently result in missing views for a subset of samples, motivating the study of *Incomplete Multi-View Clustering (IMVC)* [32].

Existing IMVC methods primarily address missing views through cross-view reconstruction, where observed views are leveraged to recover unavailable ones and facilitate subsequent clustering. Representative approaches introduce reconstruction networks that exploit correlations among views, often using complete samples as the primary source of cross-view supervision [15, 39, 4]. Comparisons indexed by nominal missing rate alone can obscure differences in the observation structure that supplies this supervision.

In this work, we show that missing rates alone are insufficient to characterize the difficulty of IMVC. Specifically, we reveal that two missing-view protocols with identical nominal missing rates may induce fundamentally different learning regimes because they contain different proportions of fully observed samples and different observation structures. We refer to this phenomenon as **protocol divergence**. Comparisons across protocols therefore need to account for realized observation structure, even when each method is retrained for each configuration.

Per-Configuration Training and Train-Once Deployment. The dominant evaluation protocol in IMVC trains and tests a separate model for each predefined combination of missing pattern and missing rate [16, 39, 4]. Such evaluation measures whether a model can adapt to a known missing distribution after retraining. However, practical deployment often follows a different scenario: a model is trained once, while the missing-view pattern may change dynamically after deployment due to varying sensor availability or environmental conditions. These two settings address fundamentally different questions. The former evaluates adaptation under retraining, whereas the latter evaluates robustness to changing missing-view configurations without additional optimization.

The Complete-Sample Supervision Bottleneck. Beyond evaluation protocols, we identify a structural limitation in reconstruction-based IMVC methods. Reconstruction objectives requiring all views rely on support-gated supervision, where a reconstruction branch is activated only when the required views are simultaneously available. Let p_c denote the proportion of fully observed samples. Although two protocols may share the same nominal missing rate, their values of p_c can differ substantially; for example, under six views and a nominal missing rate of 0.5, the complete-sample proportion may vary by approximately $50\times$ across protocols. Consequently, reconstruction-based optimization receives different amounts of effective supervision under apparently identical missing rates. We theoretically characterize this effect by deriving bounds on the gradient contribution and finite-budget optimization displacement of support-gated reconstruction objectives under explicit normalization and optimization conditions.

These observations motivate separating the supervision used to learn representations from the architecture used to process missing-view inputs at deployment. Ideally, a model should satisfy two properties: (i) for fixed shared parameters, each sample representation should depend only on its own observed views; and (ii) the fusion mechanism should explicitly account for any nonempty observed-view subset.

To achieve this goal, we propose **CRAFT** (Co-occurrence-free Robust Attention-masked Fusion Transformer), a train-once framework for incomplete multi-view clustering. CRAFT combines two architectural principles for processing missing-view inputs. First, **per-sample independence** ensures that, for fixed shared parameters, each sample representation depends on its own observed views rather than other samples' observation patterns. Second, **mask-aware variable-length fusion** aggregates only observed views for any nonempty view subset. These properties allow a shared model to process changing observed-view subsets. In the default deployment setting, CRAFT implements the "Train-Once, Deploy-Many" paradigm (Figure 3) by starting from training data with all views available and reusing one final checkpoint per dataset and seed across missing-view protocols without retraining.

Our main contributions are summarized as follows:

(1) Revealing and formalizing protocol divergence. We demonstrate that nominal missing rates fail to capture the structural difficulty of incomplete multi-view learning. We introduce effective missing rate $\hat{r}$ and complete-sample proportion p_c to quantify disparities among missing-view protocols beyond marginal missing rates.

(2) Theoretical analysis of reconstruction-based optimization. We analyze support-gated reconstruction mechanisms and establish bounds on their gradient contribution and finite-budget SGD displacement, revealing how incomplete observation structures constrain optimization.

(3) A train-once architecture for robust IMVC. We propose CRAFT, a Transformer-based framework with per-sample independence and mask-aware variable-length fusion, supporting reuse of a fixed checkpoint across changing missing-view configurations.

(4) Comprehensive empirical validation. Experiments on CUB and MultiFashion demonstrate that CRAFT achieves the strongest performance in 12 out of 13 information-matched comparisons. Further deployment experiments across seven benchmarks and sixteen missing-view configurations verify the effectiveness of checkpoint reuse and demonstrate substantial reductions in repeated-training cost.

2 Related Work

Incomplete multi-view clustering has a long pre-deep history—early methods addressed the problem through low-rank matrix factorization [14, 6, 23], incomplete multi-kernel learning [20], spectral and graph-based clustering [27, 40], and subspace alignment with missing-view inferring [30], each compensating for missing entries through a structural prior on the recovered representation. Recent tensor-completion methods additionally model cross-order neighborhoods and cross-view diversity [18].

Beyond missing-view recovery, related clustering work studies scalable single- and multi-view subspace graphs [10, 9], multi-view attributed graphs [17], high-order relations in generic multi-view data [22], and graph filtering for real-world and complex data [34, 11].

Deep IMVC methods inherit the cross-view supervision idea but train the representation end-to-end. We organize the discussion around three mechanisms concerning supervision and forward computation (Class 1, Class 2, Class 3 below). These groups are not mutually exclusive: a per-sample architecture can also be trained with support-gated reconstruction losses.

Class 1: cross-view reconstruction ($\mathcal{F}_{rec}$). The dominant paradigm trains explicit cross-view recovery modules using co-observed views as supervision. COMPLETER [15] and its TPAMI extension DCP [16] establish the template; subsequent methods refine the recovery mechanism through cognitive deep networks [31], generative recovery via GANs and diffusion [28, 39], prototype-based imputation [13], hierarchical semantic alignment [4], attention-based partial deep clustering [35], and recent Bayesian extensions [8]. The reconstruction loss is computed over pairs of views co-observed for the same sample, so this branch receives direct supervision only from samples retaining the required view pair. We denote these support-gated reconstruction losses by $\mathcal{F}_{rec}$; additional losses in the same method may use other samples.

Class 2: cross-sample and distributional methods. A second class extracts supervisory signal that is not strictly per-sample: DSIMVC [24] imputes from latent-space nearest neighbors of *other* samples, DSMVC [25] reweights views to mitigate performance degradation when adding views, SURE [37] combines cross-sample contrastive alignment with noise-robust losses for partially view-aligned and sample-missing settings, and Energy-DIMC [29] aligns view distributions through energy-based models. All four compute their training signal through cross-sample or cross-view distributional relationships rather than within-sample co-observation, placing these objectives outside the $\mathcal{F}_{rec}$ gradient bound. Their performance depends on optimization and representation quality as well as observed information.

Class 3: per-sample methods. A third class computes each sample's fused representation from its own observed views and shared parameters, with no dependence on other samples' observation patterns. DVIMC [1] uses a variational autoencoder with a Wasserstein-barycenter mixture for per-sample latent fusion; MvCLN [36], FreeCSL [3], and I2MVC [33] similarly fuse only observed views without cross-view recovery. CRAFT instantiates this per-sample forward design through attention-masked variable-length fusion, while using reconstruction and consistency objectives during representation learning. Outside clustering, attention-based fusion of observed-only modalities is established in supervised multimodal learning [21, 5], where labels provide supervision that unsupervised IMVC must instead derive from the available data.

3 Motivation and Insights

Notation. We consider n samples with up to V views and K latent clusters $Y \in \{1, \ldots, K\}$. View-v feature $x_v^{(i)} \in \mathbb{R}^{d_v}$ is observed iff $M_v^{(i)} = 1$, where $M_v^{(i)} \in \{0, 1\}$. The observed-view set is $O_i := \{v : M_v^{(i)} = 1\}$; sample i is *complete* iff $|O_i| = V$. The protocol uses nominal missing rate $r \in [0, 1]$; §3.1 gives the realized statistics on M.

3.1 Generalized Evaluations

The two protocol-induced statistics introduced in §1—the effective missing rate $\hat{r}$ and the complete-sample proportion p_c—admit closed forms over the mask matrix M:

$$\hat{r} \;=\; \frac{1}{nV} \sum_{i=1}^{n} \sum_{v=1}^{V} \big(1 - M_v^{(i)}\big), \qquad p_c \;=\; \frac{1}{n} \sum_{i=1}^{n} \prod_{v=1}^{V} M_v^{(i)}. \tag{1}$$

Both summarize the protocol's effect on the data but describe different aspects of the observation structure.

p_c describes complete-sample support. Reconstruction branches requiring all views receive supervision only from complete samples. Pairwise branches instead use samples with $|O_i| \geq 2$, whose proportion can remain positive when $p_c = 0$. The required observation support therefore depends on the loss.

$\hat{r}$ describes view coverage. Even a perfectly trained method is limited by the information available in observed views. The effective missing rate summarizes coverage but does not determine which information is retained: two protocols can share $\hat{r}$ yet differ in both p_c and the identity of the observed views. We refer to the joint $(\hat{r}, p_c)$ characterization as *protocol divergence*, since two protocols realizing the same nominal r may diverge sharply along either axis.

3.2 Four Evaluation Protocols and $50\times$ Divergence in p_c

Protocols 1 and 2 have effective missing rates $\hat{r} \leq 1/V$. Protocol 1 can still reach $p_c = 0$ at $r = 1$, when each sample retains $V - 1$ views. We introduce two complementary protocols to cover a wider range of observation structures: Protocol 3 uses per-view independent masking followed by one-view restoration for empty samples; Protocol 4 protects one view per sample and deletes a fixed total number of the remaining entries uniformly without replacement.

Three orthogonal design choices distinguish the protocols used in the IMVC literature [15, 16, 4]:

- *Granularity*: whether masking decisions are made per sample or over entries of the mask matrix;
- *Protected view*: whether one view is selected before deletion and kept throughout masking;
- *One-hot guarantee*: whether an empty sample has one view restored after masking. All four protocols here retain at least one view.

These choices yield four canonical protocols (Table 1), summarized by their limiting $\hat{r}$ and p_c as $n \to \infty$ (derivations in Appendix C.1, Monte Carlo verification in Appendix C.2):

Table 1: Four evaluation protocols and their limiting hidden variables ($\hat{r}$, p_c) as $n \to \infty$. Over the reported grid, Protocols 3–4 reach lower p_c and higher effective missing rates than Protocols 1–2. V is the total view count and r the nominal missing rate.

Protocol	Mechanism	Granularity	$\hat{r}$	p_c
1	Sample-level deletion (protected)	sample	r/V	$1 - r$
2	Entry-wise Bernoulli (protected)	entry	r/V	$\big(1 - r/(V-1)\big)^{V-1}$
3	Per-view independent (one-hot)	entry	$r - r^V/V$	$(1 - r)^V$
4	Fixed-budget deletion (protected)	entry	$\min\big(r,\, (V-1)/V\big)$	$\big[\max(0,\, 1 - Vr/(V-1))\big]^{V-1}$

What drives the divergence. The protocols combine different deletion budgets and allocation rules. Protocols 1–2 remove r/V of all entries on average, whereas Protocols 3–4 permit larger

effective missing rates; entry-wise allocation further changes the chance of retaining all views. The result, visible in Figure 1, partitions the family into a *lenient* regime (Protocols 1, 2; $p_c \geq 30\%$ throughout) and a *stringent* regime (Protocols 3, 4). For example, Protocol 1 and Protocol 4 differ by about $50\times$ in p_c at $V{=}6$, $r{=}0.5$; larger ratios occur elsewhere in the grid.

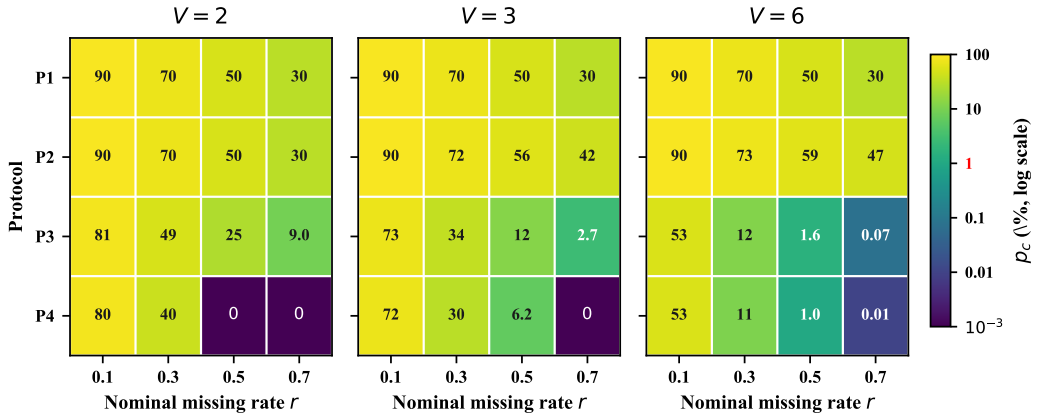


Figure 1: Complete-sample proportion p_c (%, log color scale) under the four protocols, across $V \in \{2, 3, 6\}$ and $r \in \{0.1, 0.3, 0.5, 0.7\}$. The 1% tick (red) on the colorbar provides a reference level for low complete-sample support.

Coverage gap. All nine audited IMVC implementations (Appendix C.4) concentrate on Protocols 1–2 or their variants. At $r{=}0.5$, their limiting p_c is at least 50% for $V{=}6$. This coverage gap motivates testing lower-support regimes. CRAFT is excluded from this baseline protocol audit.

4 Theoretical Analysis

We formalize two architectural conditions on the encoder f_θ that produces fused representations $h^{(i)} = f_\theta(\{x_v^{(i)}\}_{v \in O_i}) \in \mathbb{R}^d$ from observed views, and separately analyze support-gated reconstruction objectives under explicit normalization and optimization assumptions; the empirical loss is $\mathcal{L}(\theta) = \frac{1}{n} \sum_i \ell(h^{(i)}, x^{(i)})$. Full SGD-trajectory notation and theorem proofs appear in Appendix B.

Condition C1 (Per-sample independence). $h^{(i)}$ depends only on sample i's observed views and shared parameters, with no dependence on the observation patterns of other samples in its forward computation. Batch-level training regularizers are considered separately.

Condition C2 (Mask-aware variable-length fusion). f_θ is defined for any subset $O_i \subseteq \{1, \ldots, V\}$ with $|O_i| \geq 1$, producing $h^{(i)}$ via a mechanism that aggregates only observed views, including implementations using fixed slots with explicit masking.

Assumption 4.1 (Conditional independence given Y)**.** *With X_v denoting the random variable underlying $x_v^{(i)}$, $P(X_1, \ldots, X_V \mid Y) = \prod_{v=1}^{V} P(X_v \mid Y)$.*

Throughout, $\mathcal{F}_{rec}$ denotes the Class 1 reconstruction loss family of §2. We first state a universal capability bound that applies to every method regardless of architecture; the trainability results that follow are independent of this conditional-independence assumption, with Assumption 4.1 required only by the second inequality of Theorem 4.1.

Theorem 4.1 (Capability bound)**.** *For any clustering function taking a nonempty set of observed views $S \subseteq \{1, \ldots, V\}$ as input, the bound*

$$\max_{v \in S} I(X_v; Y) \;\leq\; I\big(\{X_v\}_{v \in S}; Y\big)$$

holds unconditionally. Under Assumption 4.1, the additional bound $I\big(\{X_v\}_{v \in S}; Y\big) \leq \sum_{v \in S} I(X_v; Y)$ also applies (proof in Appendix B.2).

For a learned representation, data processing gives $I(h;Y) \leq I(X_S;Y)$. The retained information depends on representation learning and optimization; this capability ceiling does not identify a method-specific failure cause.

Proposition 4.1 (Trainability bound for $\mathcal{F}_{rec}$)**.** *For a reconstruction branch normalized over all sampled examples, write its per-sample loss as $\ell_i = \mathbf{1}\{|O_i| \geq 2\}\tilde{\ell}_i$. If $\|\nabla\tilde{\ell}_i\| \leq C_g$ along the trajectory and examples are sampled uniformly, then $\mathbb{E}\|\nabla\ell_i\| \leq C_g q_2(P)$, where $q_2(P) = \mathbb{P}_{M\sim P}[|O| \geq 2]$. The same upper bound holds for the mean gradient of a uniformly sampled mini-batch. Full statement and normalization details are in Appendix B.1.*

For a strict-complete branch, replace the indicator by $\mathbf{1}\{|O_i| = V\}$ and the bound becomes $C_g p_c$. In general, $p_c \leq q_2(P)$, with equality for $V = 2$; at $V > 2$, $p_c = 0$ can coexist with $q_2(P) = 1$. At zero eligible support this branch has zero gradient, while other loss terms may still update the model.

Theorem 4.2 (Finite-budget displacement within $\mathcal{F}_{rec}$)**.** *Under Proposition 4.1's conditions, suppose constant-step SGD updates only this loss with step size η, using fresh iid samples for T steps. Then $\mathbb{E}\|\theta_T - \theta_0\| \leq \eta T C_g q_2(P)$. For strict-complete losses, replace $q_2(P)$ by p_c. The proof is in Appendix B.3.*

Role of C1+C2. C1 and C2 define a broad family of forward architectures that accept nonempty observed subsets without complete-sample input dependencies. A training objective that remains active on such subsets avoids the corresponding reconstruction support gate. These conditions alone do not guarantee successful learning.

5 CRAFT

CRAFT instantiates the two conditions of §4: C1 via per-sample forward representations, and C2 via a Transformer with attention masking [26, 21] that admits variable-length input in the spirit of Set Transformer [12] and Perceiver [7], specialized to unsupervised IMVC. These conditions define an architectural family; CRAFT is one implementation. Attention provides query-dependent weighting of observed views, while a shared cluster head applies the same clustering rule to every subset. The fusion comparison in Appendix D.11 provides empirical context for this choice.

We distinguish two tiers. **CRAFT-Core** is the reduced configuration—Stage 1 reconstruction, cosine cluster head, and attention masking. **CRAFT** (canonical) adds consistency loss, masked fine-tuning, entropy and KL regularizers, and Stage 2 fine-tuning to improve clustering, particularly at high V with few observed views (Figure 2; ablation in Appendix E.2).

5.1 Transformer Architecture

Per-view encoding. View-specific MLP encoders E_v map each view to a shared d-dimensional embedding $z_v^{(i)} = E_v(x_v^{(i)}) \in \mathbb{R}^d$. A learnable [CLS] token is prepended to the sequence of view embeddings, yielding $[[\mathrm{CLS}], z_1^{(i)}, \ldots, z_V^{(i)}]$ for each sample.

Masked self-attention. A Transformer block with multi-head self-attention is applied to the token sequence. For sample i at evaluation time with observed-view set $O_i \subseteq \{1, \ldots, V\}$, the key-padding mask is set so that positions corresponding to $v \notin O_i$ receive attention logit $-\infty$. The softmax denominator excludes these positions, ensuring missing views contribute exactly zero to the attention weights and to the resulting fused representation (Appendix B.5). The [CLS] token's output $h^{(i)}$ serves as the fused sample representation.

Cluster head. A cosine softmax classifier maps the fused representation $h^{(i)}$ to a soft cluster assignment over K clusters. The classifier weights are L_2-normalized prototypes; classification logits are scaled cosine similarities.

5.2 Two-Stage Training for Train-Once Deployment

The default deployment model starts from training data in which every sample has all V views available. Stages 1 and 2 use all views, followed by optional additional masked fine-tuning. The final

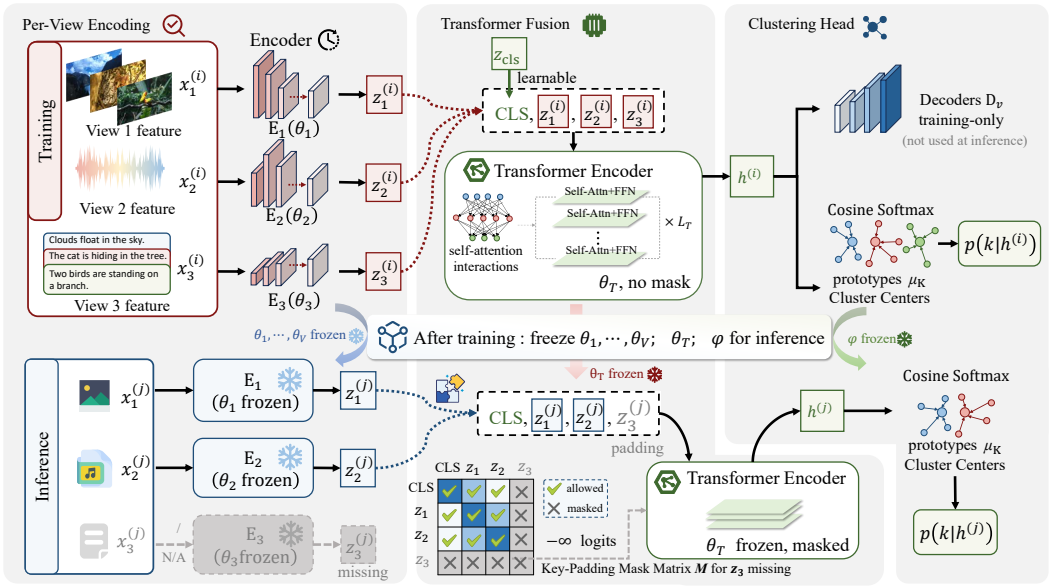


Figure 2: CRAFT architecture: per-view encoders, Transformer fusion with a learnable [CLS] token, and a cosine softmax cluster head. **Top:** default deployment training (Stages 1 and 2) with all views available; optional masked fine-tuning follows (§5.3). **Bottom:** the trained model at inference with missing views (z_3 shown missing), excluded by a key-padding mask. The depicted modalities ($V=3$, image/audio/text) and the masked-out view are illustrative only; CRAFT supports arbitrary view types and nonempty observed-view subsets at inference. Hyperparameters in Appendix A.2.

model is then frozen for evaluation under all missing-data configurations. The separate information-matched training conditions are specified in §6.1.

Stage 1: representation learning. The Stage 1 loss combines a per-view reconstruction term and a cross-view consistency term:

$$\mathcal{L}_{\text{stage1}}(\theta) \;=\; \underbrace{\frac{1}{n}\sum_{i=1}^{n}\sum_{v=1}^{V}\left\|D_v(h^{(i)}) - x_v^{(i)}\right\|^2}_{\mathcal{L}_{\text{recon}}} + \lambda_{\text{repr}}\,\mathcal{L}_{\text{repr}}(\{z_v^{(i)}\}_v), \tag{2}$$

where D_v is a view-specific decoder MLP and $\mathcal{L}_{\text{repr}}$ is a SimSiam-style consistency loss on the per-view embeddings of the same sample [2] (formal definitions: Appendix A.6). Both terms use views of the same sample, while the forward representation satisfies C1. Under this default training recipe, reconstruction and consistency provide representation-learning supervision before deployment. The frozen model subsequently processes observed views through the masked fusion described above.

Stage 2: cluster fine-tuning. The Stage 1 representation is not cluster-aligned by default: h captures multi-view structure, but its geometry need not place samples of the same cluster close to one another. Stage 2 adds a cluster head g_ϕ and a clustering objective:

$$\mathcal{L}_{\text{stage2}}(\theta,\phi) \;=\; \mathcal{L}_{\text{cluster}}(g_\phi(h^{(i)})) + \beta\,\mathcal{L}_{\text{ent}} + \gamma\,\mathcal{L}_{\text{KL}}, \tag{3}$$

Here $\mathcal{L}_{\text{cluster}}$ is a self-labeled cross-entropy on cosine softmax outputs with epoch-refreshed pseudo-labels, $\mathcal{L}_{\text{ent}}$ is an anti-collapse entropy regularizer on batch-averaged predictions, and $\mathcal{L}_{\text{KL}}$ enforces consistency between predictions from MFT-sampled view-subsets (§5.3); $\gamma=0$ during ordinary Stage 2 (formal definitions: Appendix A.6; full procedure: Algorithm 1, Appendix A.5). The entropy term aggregates predictions across a batch; the forward representation remains per-sample as specified by C1.

5.3 Masked Fine-Tuning

After Stage 2, optional *masked fine-tuning* (MFT) continues from its checkpoint for additional epochs using Eq. 3. For each sample in each batch, draw $k \sim \text{Uniform}(\{0,\ldots,V-2\})$ and

Table 2: Cross-protocol comparison. ACC (%) on three datasets. Baselines and CRAFT (per-condition) train separately per (P, r) configuration; CRAFT* (train-once) starts from training data with all views available and reuses one final checkpoint per dataset and seed across the sixteen-condition grid (Protocols 1/4 shown). HandWritten uses MFT only in the train-once row. Bold/underline = best/second per column within each dataset. † = one effective seed for the marked method or cell; unmarked CRAFT and HSACC entries are five-seed means. Other run counts follow Appendix A.1. N/A = not reported; — = runtime failure; ‡ marks ACC $\leq 40\%$.

	Protocol 1				Protocol 4			
Method	0.1	0.3	0.5	0.7	0.1	0.3	0.5	0.7
CUB (V=2, K=10)								
DCG† [39]	71.50	72.56	73.33	63.50	70.28	65.39	15.61‡	17.72‡
COMPLETER [15]	51.33	61.60	49.83	18.90‡	61.23	18.27‡	36.43‡	36.43‡
DCP [16]	65.00	57.50	51.33	26.00‡	64.17	29.67‡	36.83‡	36.83‡
HSACC [4]	63.13	54.00	43.40	29.67‡	59.50	33.13‡	38.63‡	38.63‡
Energy-DIMC [29]	75.20	67.77	58.20	51.56	71.29	53.52	<u>42.38</u>	<u>42.38</u>
DVIMC† [1]	22.67‡	27.67‡	22.67‡	22.67‡	19.83‡	26.17‡	—	—
CRAFT (per-condition)	**87.17**	<u>77.50</u>	<u>75.50</u>	<u>69.00</u>	<u>79.83</u>	<u>72.00</u>	*N/A*	*N/A*
CRAFT* (train-once)	<u>82.09</u>	**79.19**	**76.04**	**73.61**	**80.89**	**74.15**	**68.99**	**68.99**
HandWritten (V=6, K=10)								
DCG†	68.50	79.25	62.25	62.30	65.25	32.60‡	14.45‡	12.75‡
COMPLETER	92.15	67.61	89.51	75.25	85.06	89.31	58.34	12.65‡
DCP	82.35	83.90	80.00	71.55†	82.10	24.60‡	18.15‡	16.25†‡
HSACC	76.53	87.91	87.10	80.56	89.39	15.05‡	14.24‡	13.52‡
Energy-DIMC	<u>96.65</u>	<u>96.37</u>	<u>96.09</u>	<u>96.37</u>†	<u>96.09</u>	<u>94.87</u>	<u>92.75</u>	<u>83.26</u>†
DVIMC†	37.30‡	84.05	79.55	24.30‡	71.40	20.00‡	77.05	—
CRAFT (per-condition)	94.75	94.59†	94.99	95.03	94.92	93.98	91.05	82.56
CRAFT* (train-once)	**97.60**	**97.57**	**97.57**	**97.58**	**97.53**	**96.82**	**93.69**	**85.21**
MultiFashion (V=3, K=10)								
DCG†	96.29	91.99	85.65	76.25	91.70	71.78	48.22	11.68‡
COMPLETER	79.92	87.23	90.66	<u>90.09</u>	83.08	87.87	46.37	—
DCP	86.04	72.75	72.26	68.95	75.24	67.21	69.36	25.11‡
HSACC	<u>97.46</u>	<u>96.40</u>	91.67	83.36	<u>94.34</u>	82.50	49.80	21.61‡
Energy-DIMC	93.30	91.64	90.19	89.90	91.97	88.53	79.77	<u>29.14</u>‡
DVIMC	83.86	86.62	88.82	87.21	86.65	84.09	80.43	—
CRAFT (per-condition)	**98.07**	**97.32**	**96.69**	89.61	**97.35**	**94.21**	<u>86.59</u>	*N/A*
CRAFT* (train-once)	93.21	92.82	<u>92.44</u>	**91.97**	92.73	<u>90.83</u>	**88.12**	**85.41**

randomly exclude k views via the same key-padding mask used at inference. This exposes the model to observed-view subsets through the same masking mechanism used at inference and acts as a view-subset ensemble (Appendix B.6). For $V \geq 3$, the rule can sample multiple subsets retaining at least two views; for $V = 2$, only the full-view subset contributes, so this sampling rule performs no view deletion. Any feature noise is a separate augmentation.

6 Experiments

Our experiments answer four questions in turn: (i) how does CRAFT compare under matched incomplete training? (ii) how do observation support and the choice of recovery loss relate to baseline behavior? (iii) how well does a single final CRAFT checkpoint cover the deployment grid, and at what cost? (iv) how do individual CRAFT components affect clustering performance? Cross-protocol comparison (§6.2) addresses (i)–(iii); the component ablation (§6.3) addresses (iv).

6.1 Setup

Baselines and protocol grid. We evaluate CRAFT against the three classes of §2: Class 1 (COMPLETER [15], DCP [16], DCG [39], HSACC [4]), Class 2 (Energy-DIMC [29]), and Class 3 (DVIMC [1]), across the protocol-rate grid of §3.2. Three datasets are reported in the main text; four more, along with hyperparameters, hardware, seed protocol and metric details, appear in Appendix A.1. The deployment tables use the canonical configuration of §5.

Training and evaluation conditions. The information-matched CUB and MultiFashion comparison trains each method separately for each listed (P, r) condition, using matched observed training information, preprocessing, model-selection information, and evaluation masks. CRAFT uses observed inputs and targets, with consistency over eligible pairs (Appendix A.2). Separately, the train-once deployment grid starts CRAFT from training data with all views available and reuses one final checkpoint per dataset and seed across all sixteen configurations; baselines retrain per condition. The HandWritten rate-matched results are included in Table 2, with a comparison disabling MFT in both variants in Appendix D.6. Independent-run counts and implementation exceptions are given in Appendix A.1.

6.2 Cross-Protocol Comparison

CRAFT leads most information-matched conditions. Table 2 places CRAFT (per-condition) first in all six CUB conditions and six of seven MultiFashion conditions. The exception is MultiFashion under Protocol 1 at $r = 0.7$, where COMPLETER reaches 90.09%, followed by Energy-DIMC at 89.90% and CRAFT at 89.61%.

Observation support depends on the reconstruction loss. The HandWritten configuration using Protocol 1 at $r = 1$ has $p_c = 0$ while every sample retains five views. Complete-sample reconstruction is inactive, whereas pairwise reconstruction retains support (Appendix D.9).

A single final CRAFT checkpoint covers all sixteen deployment configurations. Table 2 shows the Protocol 1 and Protocol 4 portions of this grid; the full results are in Appendix D. CRAFT* exceeds every baseline in the displayed CUB and HandWritten conditions, while some MultiFashion conditions favor per-configuration baselines. The HandWritten rate-matched control differs from its no-MFT reference trained with all views available by at most 2.91 points over the reported conditions (Appendix D.6).

Training cost and scale. The recorded MultiFashion costs are 128 minutes for a sixteen-run DVIMC grid and 14.5 minutes for one CRAFT run, a ratio of $8.8\times$; these records do not itemize model selection or the additional MFT stage (Appendix D.8). For embedding dimension d and L Transformer layers, fixed-slot fusion takes $O(L(V+1)^2 d + L(V+1)d^2)$ time per sample, in addition to the view encoders/decoders; attention storage is $O(L(V+1)^2)$ for a fixed head count. Training is linear in the number of samples per epoch for fixed widths and view count. YTF-31, with $101{,}499$ samples, supplies the largest tested scale, using a single seed.

6.3 Component Ablation

The reconstruction loss is central to Stage 1; Stage 2 effects vary by dataset. A subtractive ablation under Protocol 4 yields three qualitative findings; per-rate numbers appear in Appendix E.1. These three-seed ablations use the original learnable-placeholder recipe and their within-table Full reference. Consistency-only training performs poorly on HandWritten, while removing reconstruction also lowers CUB accuracy. Removing Stage 2 or entropy changes HandWritten accuracy by about one point, with larger changes on CUB. Consistency and masked fine-tuning have dataset-dependent effects; the CUB reconstruction-only variant improves on its Full reference.

CRAFT-Core preserves useful accuracy with a reduced training recipe. CRAFT-Core exceeds the listed reconstruction baselines on CUB, but on HandWritten it does not win every configuration. The gap to canonical CRAFT grows at high missing rates (Appendix E.2); this comparison removes several components together and cannot identify MFT's isolated effect. A four-hyperparameter sensitivity sweep (Appendix E.3) supplies the tested parameter ranges.

Attention fusion outperforms the tested pooling and concat blocks. Table 3 compares fusion architectures under Protocol 4; Appendix D.11 reports all four protocols. The variants share the

encoders and two-stage training without MFT, and differ in the fusion block. The Transformer leads both SetMLP widths and concat fusion at every reported rate. Increasing the SetMLP width does not close this gap in the tested configurations. This within-table comparison supports the fusion choice; its no-MFT recipe is separate from canonical CRAFT in Table 2.

Table 3: Fusion comparison on HandWritten under Protocol 4. ACC (%), five-seed mean ± std. All variants share the encoders and two-stage training without MFT, differing in their fusion block. Bold = best per column. The full four-protocol grid is in Appendix D.11.

Fusion	$r = 0.1$	$r = 0.3$	$r = 0.5$	$r = 0.7$
Transformer + attn mask	**97.14**±1.1	**95.59**±1.5	**90.79**±3.0	**79.06**±5.7
SetMLP-Light ($h' = 192$)	94.01±1.2	88.82±1.3	78.97±1.8	62.09±2.3
SetMLP-Match ($h' = 560$)	92.93±1.7	85.89±1.7	73.83±1.7	55.15±1.9
Concat fusion	93.38±0.4	86.89±0.2	75.04±0.4	56.55±0.7

Excluding missing tokens improves the fixed-checkpoint comparison. Table 4 compares four inference-time treatments of missing views using the same checkpoint within each run, trained with all views available. Attention masking has the highest mean ACC at all three reported rates. The alternative strategies retain missing positions in attention, whereas the mask excludes their contribution exactly (Appendix B.5). These results isolate inference-time handling within this comparison, rather than a change in training.

Table 4: Missing-view strategy comparison on CUB under Protocol 4. ACC (%), three-seed mean ± std. The strategies use the same checkpoint within each run, trained with all views available, differing only in inference-time handling of missing-view tokens. Bold = best per row.

r	Attn Mask	Learnable	Zero	Mean
0.1	**77.29**±4.1	76.86±3.7	76.36±3.4	75.77±2.6
0.3	**69.78**±5.4	68.62±4.4	66.40±3.3	65.03±0.2
0.5	**63.84**±7.9	61.54±5.7	58.13±4.4	55.89±0.9

7 Conclusion

We introduced CRAFT, which combines representation learning with per-sample, mask-aware fusion for deployment under changing observed-view subsets. Our analysis bounds the contribution of support-gated reconstruction branches, distinguishing complete-sample from pairwise observation support. By combining per-sample representations and mask-aware fusion, CRAFT handles changing observed subsets with a shared clustering rule. The information-matched CUB and MultiFashion comparison places CRAFT first in twelve of thirteen conditions. Separately, deployment training starts from data with all views available and yields one final checkpoint per dataset and seed for the deployment grid, with lower recorded cumulative training cost. Our work suggests that IMVC research should report observation structure alongside nominal missing rate and evaluate both incomplete training and model reuse. The four controlled protocols expose structural differences in synthetic missingness; naturally occurring, quality-dependent and correlated sensor failures remain untested. Training when an entire modality is never observed, and strictly zero-complete-sample training beyond the reported conditions, require further study.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (No. U24A20323). We acknowledge support from the Center for HPC, University of Electronic Science and Technology of China.

References

[1] Wenlan Chen, Lu Gao, Cheng Liang, and Fei Guo. Deep variational incomplete multi-view clustering with information-theoretic guidance. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25)*, pages 2457–2466. ACM, 2025.

[2] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15750–15758, 2021.

[3] Yuzhuo Dai, Jiaqi Jin, Zhibin Dong, Siwei Wang, Xinwang Liu, En Zhu, Xihong Yang, Xinbiao Gan, and Yu Feng. Imputation-free and alignment-free: Incomplete multi-view clustering driven by consensus semantic learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5071–5081, 2025.

[4] Xiaojian Ding, Lin Zhao, Xian Li, and Xiaoying Zhu. Incomplete multi-view clustering via hierarchical semantic alignment and cooperative completion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[5] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, 2022.

[6] Menglei Hu and Songcan Chen. Doubly aligned incomplete multi-view clustering. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2262–2268, 2018.

[7] Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver: General perception with iterative attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4651–4664. PMLR, 2021.

[8] Jiaqi Jin, Siwei Wang, Zhibin Dong, Xihong Yang, Xinwang Liu, En Zhu, and Kunlun He. Deep incomplete multi-view clustering with distribution dual-consistency recovery guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1016–1026, 2025.

[9] Zhao Kang, Wangtao Zhou, Zhitong Zhao, Junming Shao, Meng Han, and Zenglin Xu. Large-scale multi-view subspace clustering in linear time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4412–4419, 2020. doi: 10.1609/aaai.v34i04.5867.

[10] Zhao Kang, Zhiping Lin, Xiaofeng Zhu, and Wenbo Xu. Structured graph learning for scalable subspace clustering: From single view to multiview. *IEEE Transactions on Cybernetics*, 52 (9):8976–8986, 2022. doi: 10.1109/TCYB.2021.3061660.

[11] Zhao Kang, Xuanting Xie, Bingheng Li, and Erlin Pan. CDC: A simple framework for complex data clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 36(7):13177–13188, 2025. doi: 10.1109/TNNLS.2024.3473618.

[12] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning (ICML)*, pages 3744–3753, 2019.

[13] Haobin Li, Yunfan Li, Mouxing Yang, Peng Hu, Dezhong Peng, and Xi Peng. Incomplete multi-view clustering via prototype-based imputation. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3911–3919, 2023.

[14] Shao-Yuan Li, Yuan Jiang, and Zhi-Hua Zhou. Partial multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1968–1974, 2014.

[15] Yijie Lin, Yuanbiao Gou, Zitao Liu, Boyun Li, Jiancheng Lv, and Xi Peng. COMPLETER: Incomplete multi-view clustering via contrastive prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11174–11183, 2021.

[16] Yijie Lin, Yuanbiao Gou, Xiaotian Liu, Jinfeng Bai, Jiancheng Lv, and Xi Peng. Dual contrastive prediction for incomplete multi-view representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4447–4461, 2023.

[17] Zhiping Lin, Zhao Kang, Lizong Zhang, and Ling Tian. Multi-view attributed graph clustering. *IEEE Transactions on Knowledge and Data Engineering*, 35(2):1872–1880, 2023. doi: 10.1109/TKDE.2021.3101227.

[18] Chundan Liu, Yongyong Chen, Zhao Kang, Junyu Dong, Panpan Zheng, Chong Peng, and Qiang Cheng. Incomplete multi-view clustering via cross-view diversity enhanced fine-grained tensor completion. *IEEE Transactions on Knowledge and Data Engineering*, 2026. doi: 10.1109/TKDE.2026.3710602. In press.

[19] Ruimeng Liu, Xin Zou, Chang Tang, Xiao Zheng, Xingchen Hu, Kun Sun, and Xinwang Liu. SparseMVC: Probing cross-view sparsity variations for multi-view clustering. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[20] Xinwang Liu, Miaomiao Li, Lei Wang, Yong Dou, Jianping Yin, and En Zhu. Multiple kernel k-means with incomplete kernels. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2259–2265, 2017.

[21] Mengmeng Ma, Jian Ren, Long Zhao, Davide Testuggine, and Xi Peng. Are multimodal transformers robust to missing modality? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18177–18186, 2022.

[22] Erlin Pan and Zhao Kang. High-order multi-view clustering for generic data. *Information Fusion*, 100:101947, 2023. doi: 10.1016/j.inffus.2023.101947.

[23] Weixiang Shao, Lifang He, and Philip S. Yu. Multiple incomplete views clustering via weighted nonnegative matrix factorization with $l_{2,1}$ regularization. In *ECML/PKDD*, 2015.

[24] Huayi Tang and Yong Liu. Deep safe incomplete multi-view clustering: Theorem and algorithm. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, pages 21090–21110, 2022.

[25] Huayi Tang and Yong Liu. Deep safe multi-view clustering: Reducing the risk of clustering performance degradation caused by view increase. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 202–211, 2022.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6000–6010, 2017.

[27] Hao Wang, Linlin Zong, Bing Liu, Yan Yang, and Wei Zhou. Spectral perturbation meets incomplete multi-view data. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3677–3683, 2019.

[28] Qianqian Wang, Zhengming Ding, Zhiqiang Tao, Quanxue Gao, and Yun Fu. Generative partial multi-view clustering with adaptive fusion and cycle consistency. *IEEE Transactions on Image Processing*, 30:1771–1783, 2021. doi: 10.1109/TIP.2020.3048626.

[29] Ziyu Wang, Yiming Du, Rui Ning, and Lusi Li. Energy-based deep incomplete multi-view clustering. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25)*, pages 1686–1694. ACM, 2025.

[30] Jie Wen, Zheng Zhang, Yong Xu, Bob Zhang, Lunke Fei, and Hong Liu. Unified embedding alignment with missing views inferring for incomplete multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019. doi: 10.1609/aaai.v33i01.33015393.

[31] Jie Wen, Zheng Zhang, Yong Xu, Bob Zhang, Lunke Fei, and Guo-Sen Xie. CDIMC-net: Cognitive deep incomplete multi-view clustering network. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3230–3236, 2020. doi: 10.24963/ijcai.2020/447.

[32] Jie Wen, Zheng Zhang, Lunke Fei, Bob Zhang, Yong Xu, Zhao Zhang, and Jinxing Li. A survey on incomplete multiview clustering. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(2):1136–1149, 2022.

[33] Benyu Wu, Wei Du, Jun Wang, and Guoxian Yu. Imputation-free incomplete multi-view clustering via knowledge distillation. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 6570–6578, 2025.

[34] Xuanting Xie, Erlin Pan, Zhao Kang, Wenyu Chen, and Bingheng Li. Provable filter for real-world graph clustering. *IEEE Transactions on Cybernetics*, 56(9):5171–5184, 2026. doi: 10.1109/TCYB.2026.3671252.

[35] Jie Xu, Chao Li, Liang Peng, Yazhou Ren, Xiaoshuang Shi, Heng Tao Shen, and Xiaofeng Zhu. Adaptive feature projection with distribution alignment for deep incomplete multi-view clustering. *IEEE Transactions on Image Processing*, 32:1354–1366, 2023.

[36] Mouxing Yang, Yunfan Li, Zhenyu Huang, Zitao Liu, Peng Hu, and Xi Peng. Partially view-aligned representation learning with noise-robust contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1134–1143, 2021.

[37] Mouxing Yang, Yunfan Li, Peng Hu, Jinfeng Bai, Jiancheng Lv, and Xi Peng. Robust multi-view clustering with incomplete information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):1055–1069, 2023. doi: 10.1109/TPAMI.2022.3155499.

[38] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3391–3401, 2017.

[39] Yuanyang Zhang, Yijie Lin, Weiqing Yan, Li Yao, Xinhang Wan, Guangyuan Li, Chao Zhang, Guanzhou Ke, and Jie Xu. Incomplete multi-view clustering via diffusion contrastive generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22650–22658, 2025.

[40] Handong Zhao, Hongfu Liu, and Yun Fu. Incomplete multi-modal visual data grouping. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2392–2398, 2016.

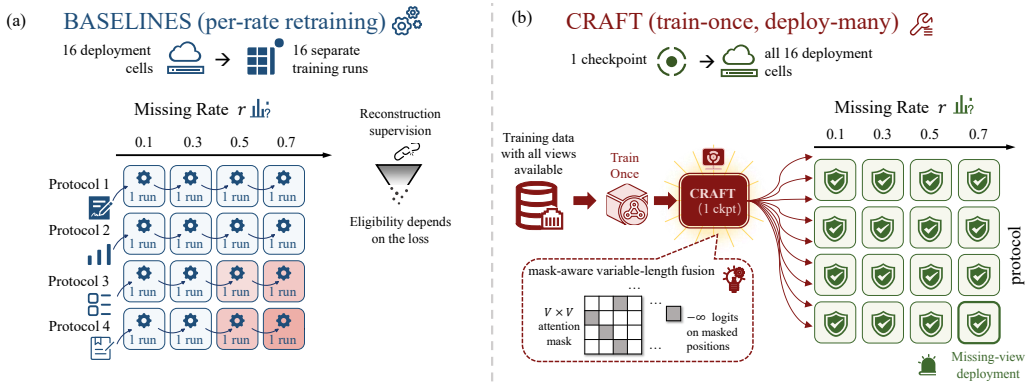


Figure 3: Per-configuration retraining vs. train-once deployment. **(a)** Baselines retrain a separate model per $(\text{protocol}, r)$ configuration; reconstruction supervision depends on the observations required by the loss. **(b)** CRAFT starts from training data with all views available and reuses one final checkpoint per dataset and seed across all 16 deployment configurations (including $p_c \to 0$) via attention masking, replacing sixteen separate training runs.

A Experimental Details

A.1 Datasets and Hardware

Table 5 summarizes the seven benchmarks used in this work.

Table 5: Dataset overview.

Dataset	N	V	K	d_{total}
CUB	600	2	10	1,324
MultiFashion	10,000	3	10	1,176
UCI-digit	2,000	3	10	356
Out-Scene	2,688	4	8	1,248
HandWritten	2,000	6	10	649
Caltech	2,386	6	20	3,766
YTF-31	101,499	5	31	2,125

All experiments are conducted on a single NVIDIA RTX 3090 GPU (24 GB VRAM) with PyTorch 2.1 and CUDA 12.1. Timing records refer to this hardware. The information-matched comparison reports five-seed means. In the historical deployment tables, CRAFT and HSACC use five seeds (42–46), except YTF-31 (seed 42). DCG's fixed-seed repetition has one effective trajectory; DVIMC on CUB and HandWritten is single-seed. The four Energy-DIMC HandWritten entries at $r = 0.7$ and the unannotated DCP/DCG entries in the support comparison are single-seed. Other baseline aggregates without an established independent-run count carry no five-seed claim. Ablations use three seeds where stated. Standard deviations are displayed only where supplied; these comparisons do not establish statistical significance.

A.2 CRAFT Hyperparameters

Table 6 lists the reference per-dataset configuration for deployment training. All datasets use a single-layer Transformer with 4 heads, dataset-specific MLP encoders and symmetric decoders, and the same optimizer (AdamW, weight decay 10^{-4}, batch size 256). For deployment, configuration selection is performed once using training data with all views available, and the final checkpoint after the selected training stages handles all missing patterns without further adjustment. The CUB deployment results specify $\lambda_{\text{repr}} = 0.1$ and $\tau = 0.5$ in Appendix D.1. The retained records do not establish the checkpoint-selection criterion for every historical run.

CUB uses d=128 because the small sample size (n=600) leads to overfitting at d=256. Caltech and MultiFashion use shallow encoders ($d_v \to d$) because their high-dimensional views cause train-

Table 6: Per-dataset hyperparameter configuration. "Default" denotes the shared configuration used by HW and UCI without modification.

Parameter	Default	CUB	MF	OS	Cal	YTF
encoder_type	deep	deep	shallow	deep	shallow	deep
encoder_hidden	512	512	1024	512	512	1024
embed_dim d	256	128	256	256	256	512
τ	0.5	0.5	0.03	0.1	0.5	0.5
λ_{ent}	5.0	5.0	0.5	5.0	0.1	0.1
dropout	0.1	0.1	0.03	0.1	0.1	0.1
noise_std	0.1	0.1	0.03	0.1	0.1	0.1
Stage 1 epochs	100	100	200	100	100	100
Stage 2 epochs	100	100	200	100	100	80
Stage 1 lr	5e-4	5e-4	5e-4	5e-4	5e-4	5e-4
Stage 2 lr	1e-5	1e-5	5e-5	1e-4	2e-4	1e-6

ing instability with deeper encoders. YTF-31 uses a wider architecture (d=512, hidden=1024) to accommodate its larger scale and higher class count (K=31). MFT resumes from the Stage 2 checkpoint for an additional 100 epochs on HW and Out-Scene or 300 epochs on MultiFashion, with learning rate 10^{-5}. These epochs are not included in the Stage 2 budgets in Table 6.

The information-matched comparison trains CUB at Protocol 1 $r \in \{0.1, 0.3, 0.5, 0.7\}$ and Protocol 4 $r \in \{0.1, 0.3\}$, and MultiFashion at Protocol 1 $r \in \{0.1, 0.3, 0.5, 0.7\}$ and Protocol 4 $r \in \{0.1, 0.3, 0.5\}$. Each condition uses its own training run. Encoders and reconstruction targets use only observed entries; pairwise consistency is averaged over eligible observed pairs and is zero for singleton samples. Preprocessing and model selection use the same training-information budget across compared methods. The separate deployment hyperparameters above describe training that starts with all views available.

A.3 Baseline Reproduction

This section consolidates reproduction details, protocol inferences, and failure analyses for all baseline methods.

A.3.1 DCG Configuration

The official DCG repository contains no multi-view configuration beyond two views. We extend the released two-view script to accept all views, retaining all other hyperparameters at their official values.

A.3.2 View Selection Audit

On HandWritten (V=6, n=2,000), the official DCG and ProImp codebases load only 2 of the 6 views: DCG selects views 2 and 3 (76d and 64d), discarding 78% of the total feature dimensions—a choice not documented in the corresponding paper. ProImp's encoder pair is hardcoded to 2 views by construction. The DCG numbers reported in this paper instead use our 6-view extension (Appendix A.3.1), which accesses all six views while retaining DCG's other hyperparameters at their official values. ProImp and APADC are architecturally limited to two views and are not extended in this work.

A.3.3 DCP and DSIMVC Implementation

DCP. DCP shares its masking utility with COMPLETER (same research group and codebase); its behavior matches Protocol 2 (§3.2).

DSIMVC. Generates masks by drawing a per-sample Bernoulli trial at rate r, then deleting one randomly chosen view for each selected sample. A post-processing step ensures at least one view per sample. This is equivalent to Protocol 1 with Bernoulli rather than fixed-count sampling; the difference is negligible for large n.

Evaluation randomness. In DCG, the outer-loop variable intended to vary across repeated runs is not connected to the random seed; a fixed seed produces the same mask and training trajectory in every iteration, reducing the effective number of independent trials to one. In ProImp, the mask seed varies but the training seed is fixed. Repeating an identical trajectory does not add an independent trial. DCG is therefore marked single-seed in the deployment tables; the information-matched table has its own stated aggregation.

A.3.4 BURG Protocol Inference

BURG [8] does not specify its masking procedure. We classify it as Protocol 2 by convention: BURG benchmarks against COMPLETER, DCP, and APADC at identical nominal rates (MR $\in \{0.1, 0.3, 0.5, 0.7\}$) without discussing protocol differences, and our audit (§C.4) verified by code inspection that these three baselines realize Protocol 2 via a shared entry-wise Bernoulli with protected-view masking codebase. This suggests BURG inherits the same masking convention, though the classification is not directly verifiable from the paper alone.

A.3.5 DVIMC Failure on Multi-View Datasets

DVIMC reports initialization `ValueErrors` on MultiFashion at $r = 0.7$ (Protocol 4), CUB at $r \geq 0.5$ (Protocol 4), and HandWritten under Protocols 3–4 at $r = 0.7$. Protocols 3 and 4 retain at least one observed view per sample, so these failures cannot be attributed to empty observed sets under the stated masks. The retained records establish the error category, but not a common root cause. COMPLETER's failed MultiFashion/Protocol 4/$r = 0.7$ entry records a data-loader runtime error. These failed entries are excluded from numeric ranking.

Single-seed evaluation of DVIMC on CUB and HandWritten also shows low and non-monotone ACC across successful configurations. These observations concern the evaluated implementation and hyperparameters, and do not establish a failure of all per-sample architectures or the necessity of C2.

Under the identical 16 configurations, the canonical CRAFT checkpoint produces non-trivial accuracy on every condition with no crashes (Table 10).

A.4 Tuning of Collapsed Baselines

To test sensitivity to selected optimization settings, we tuned DCG (lr$\in$ {1e-3, 5e-4, 1e-4}, $\tau \in$ {0.5, 0.3, 0.1}) and DCP (lr$\in$ {1e-3, 5e-4, 1e-4}) under Protocol 4 on CUB (V=2, r=0.5, p_c=0). No DCG configuration exceeded 20.17% ACC and no DCP configuration exceeded 36.83% (random baseline 10%). These searches did not recover high accuracy within the tested ranges; they do not exclude all optimization or implementation effects.

A.5 Training Algorithm

Algorithm 1 CRAFT training for default train-once deployment. Pseudo-labels are refreshed at the start of each Stage 2 and MFT epoch.

Require: complete samples $\{x^{(i)}\}_{i=1}^{n}$; encoder θ, head ϕ; epochs $E_1, E_2, E_{\mathrm{MFT}}$ ($E_{\mathrm{MFT}} = 0$ if MFT is disabled); coefficients $\lambda_{\mathrm{repr}}, \beta, \gamma$

for each minibatch in E_1 epochs {Stage 1: representation learning} **do**
  Compute $z_v = E_v(x_v)$ for $v = 1{:}V$
  $h \leftarrow \mathrm{Transformer}([[\mathrm{CLS}], z_1, \ldots, z_V])$ (no mask)
  Update θ on $\mathcal{L}_{\mathrm{stage1}}$ (Eq. 2)
end for
for each minibatch in E_2 epochs {Stage 2: cluster fine-tuning} **do**
  Compute $z_v = E_v(x_v)$ for $v = 1{:}V$
  $h \leftarrow \mathrm{Transformer}([[\mathrm{CLS}], z_1, \ldots, z_V])$ (no mask)
  Update θ, ϕ on $\mathcal{L}_{\mathrm{stage2}}$ with $\gamma = 0$ (Eq. 3)
end for
for each minibatch in E_{MFT} additional epochs {Optional MFT continuation} **do**
  Compute $z_v = E_v(x_v)$; sample two MFT view-subsets S_1, S_2
  Compute h_{S_1}, h_{S_2} using their respective key-padding masks
  Update θ, ϕ on $\mathcal{L}_{\mathrm{stage2}}$, including MFT consistency (Eq. 3)
end for
Inference (nonempty observed-view subset): forward pass with key-padding mask set from O_i.

A.6 Loss Component Definitions

For the default deployment-training recipe, the Stage 1 and Stage 2 losses (Eqs. 2–3) use the following terms; entropy aggregates predictions across samples.

$\mathcal{L}_{\mathrm{recon}}$ **(per-view reconstruction).** $\mathcal{L}_{\mathrm{recon}} = \frac{1}{n}\sum_{i=1}^{n}\sum_{v=1}^{V}\big\|D_v(h^{(i)}) - x_v^{(i)}\big\|^2$, where D_v is a view-specific decoder MLP.

$\mathcal{L}_{\mathrm{repr}}$ **(cross-view consistency).** $\mathcal{L}_{\mathrm{repr}} = -\frac{1}{nV(V-1)}\sum_{i}\sum_{v\neq v'}\cos\big(P_\phi(z_v^{(i)}),\, \mathrm{sg}(z_{v'}^{(i)})\big)$, with predictor MLP P_ϕ and stop-gradient $\mathrm{sg}(\cdot)$ [2].

$\mathcal{L}_{\mathrm{cluster}}$ **(self-labeled cross-entropy).** $\mathcal{L}_{\mathrm{cluster}} = -\frac{1}{n}\sum_{i=1}^{n}\log p_{\hat{y}^{(i)}}\big(h^{(i)}\big)$, $p_k(h) = \mathrm{softmax}_k\big(\tau\cos(h, c_k)\big)$, where $\hat{y}^{(i)} = \arg\max_k p_k(h^{(i)})$ are pseudo-labels refreshed each epoch and $\{c_k\}_{k=1}^{K}$ are unit-norm prototypes.

$\mathcal{L}_{\mathrm{ent}}$ **(anti-collapse entropy).** $\mathcal{L}_{\mathrm{ent}} = \sum_{k=1}^{K}\bar{p}_k\log\bar{p}_k$, $\bar{p}_k = \frac{1}{n}\sum_{i=1}^{n}p_k\big(h^{(i)}\big)$. Minimizing this term pushes the batch-averaged prediction toward uniform over K clusters, discouraging degenerate single-cluster solutions.

$\mathcal{L}_{\mathrm{KL}}$ **(MFT consistency).** For two MFT-sampled view-subsets $S_1, S_2 \subseteq O_i$ of sample i (§5.3), $\mathcal{L}_{\mathrm{KL}} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{KL}\big(p(\cdot \mid h_{S_1}^{(i)}) \,\big\|\, p(\cdot \mid h_{S_2}^{(i)})\big)$, with $\gamma = 0$ when MFT is disabled.

A.7 PyTorch Inference Snippet

The following snippet illustrates CRAFT's inference-time handling of missing views. No special module is required; the standard PyTorch `MultiheadAttention` accepts a key-padding mask directly.

```
# tokens: [batch, 1+V, d]  (CLS + V view embeddings)
# missing: [batch, V] bool  (True = view absent)
pad = torch.zeros(B, 1+V, dtype=torch.bool)  # CLS never masked
pad[:, 1:] = missing
h = transformer_encoder(tokens, src_key_padding_mask=pad)
h_cls = h[:, 0]  # fused representation
pred = cosine_softmax(h_cls, centroids, tau)
```

B Proofs

B.1 Proof of Proposition 4.1 (Trainability Bound for $\mathcal{F}_{rec}$)

Proof of Proposition 4.1. For a pairwise reconstruction branch, let $\ell_i = \mathbf{1}\{|O_i| \geq 2\}\tilde{\ell}_i$, with $\|\nabla\tilde{\ell}_i\| \leq C_g$. The loss is normalized by the total sample count, including ineligible samples with zero loss. Then

$$\mathbb{E}\|\nabla\ell_i\| \leq C_g\,\mathbb{P}(|O_i| \geq 2) = C_g q_2(P).$$

For a uniformly sampled batch of size B, the averaged gradient $g = B^{-1}\sum_{i=1}^{B}\nabla\ell_i$ satisfies $\mathbb{E}\|g\| \leq B^{-1}\sum_i \mathbb{E}\|\nabla\ell_i\| \leq C_g q_2(P)$ by the triangle inequality. A strict-complete branch instead uses $\mathbf{1}\{|O_i| = V\}$, giving $C_g p_c$. For a loss associated with one specific view pair, use that pair's co-observation probability; q_2 remains an upper envelope. □

Normalization and active batches. The same support can yield a different bound when gradients are averaged over eligible samples only. For a strict-complete branch, let $B_c \sim \mathrm{Binomial}(B, p_c)$ under iid mask sampling, and define

$$g_t = \begin{cases} B_c^{-1}\sum_{i\in C_{\mathrm{batch}}}\nabla\tilde{\ell}_i, & B_c \geq 1, \\ 0, & B_c = 0. \end{cases}$$

Here C_{batch} is the set of complete samples in the batch. Then $\mathbb{E}\|g_t\| \leq C_g[1-(1-p_c)^B]$, and the expected active-batch size is $\mathbb{E}[B_c \mid B_c \geq 1] = Bp_c/[1-(1-p_c)^B]$ for $p_c > 0$. For $B = 256$, the active-batch probability rises from 10% to 90% between $p_c \approx 0.04\%$ and 0.90%. These are batch-availability levels, not accuracy thresholds. Uniform resampling of a fixed masked dataset uses its empirical complete fraction; eligibility-stratified sampling changes this law. Neither this counting argument nor a standard convergence upper bound establishes a necessary $1/p_c$ iteration lower bound.

Remark (fixed-topology failure, ProImp case). The `cross_rec()` module of ProImp constructs tensors of shape $|C| \times K$ and $|C| \times |C|$, where C is the complete-sample set. Under Protocol 4 with $V = 2$ and $r \geq 0.5$, $|C| = 0$. The recorded fallback returns incompatible tensor shapes, triggering `RuntimeError: The size of tensor a (1) must match the size of tensor b (600)`. This is an implementation failure at empty complete-sample support.

Remark (CRAFT forward computation). For fixed parameters, sample i's representation uses its observed features and the key-padding mask. The fixed-slot implementation retains $V+1$ tokens, including CLS, and masks missing keys; its matrix dimensions do not shrink with $|O_i|$. The forward map remains defined for any nonempty O_i, including deployment configurations with $p_c = 0$. This property is separate from the training-information conditions of each experiment.

B.2 Capability Bound: Theorem Statement and Proof

This subsection states and proves the capability bound referenced throughout the main text (§4, §6.2, §7). The bound limits the cluster-relevant information any clustering function can extract from

a given observed-view subset, regardless of training dynamics or architecture, and applies to all methods including those outside $\mathcal{F}_{rec}$ (e.g., distributional methods such as Energy-DIMC).

We prove Theorem 4.1 (statement in §4). The upper inequality requires conditional independence and finite information quantities; without that assumption, the lower inequality and data-processing upper bound still hold.

Proof of Theorem 4.1. **Upper bound.** Under Assumption 4.1 (conditional independence of views given Y), the joint entropy decomposes as $H(\{X_v\}_{v\in O_i} \mid Y) = \sum_{v\in O_i} H(X_v \mid Y)$. By the subadditivity of entropy, $H(\{X_v\}_{v\in O_i}) \le \sum_{v\in O_i} H(X_v)$. Therefore,

$$\begin{aligned} I(\{X_v\}_{v\in O_i}; Y) &= H(\{X_v\}_{v\in O_i}) - H(\{X_v\}_{v\in O_i} \mid Y) \\ &\le \sum_{v\in O_i} H(X_v) - \sum_{v\in O_i} H(X_v \mid Y) \\ &= \sum_{v\in O_i} I(X_v; Y). \end{aligned} \tag{4}$$

Lower bound. For any $u \in O_i$, X_u is a component of $\{X_v\}_{v\in O_i}$. Adding more variables cannot decrease mutual information: $I(\{X_v\}_{v\in O_i}; Y) \ge I(X_u; Y)$. Taking the maximum over $u \in O_i$ gives the lower bound.

Application to fused representations. Since the fused representation $\mathbf{h}_{\text{cls}}^{(i)}$ is a deterministic function of $\{X_v\}_{v\in O_i}$, the data processing inequality gives $I(\mathbf{h}_{\text{cls}}^{(i)}; Y) \le I(\{X_v\}_{v\in O_i}; Y)$. The lower bound on the observed set does not transfer to the learned representation: a constant encoder can discard all cluster information. When $|O_i| = 1$, the observed-set information is that of the single retained view; architecture and optimization still affect how much of it is preserved. These statements apply to fixed encoders on the evaluation distribution, with any training information treated as fixed. □

B.3 Proof of Theorem 4.2 (Finite-Budget Displacement within $\mathcal{F}_{rec}$)

This appendix proves Theorem 4.2 of the main text using the gradient-counting and parameter-displacement steps below. The result concerns a support-gated reconstruction loss under SGD.

B.3.1 Notation and Setting

Let $\Theta \subseteq \mathbb{R}^p$ be the parameter space and $\theta_0 \in \Theta$ the initialization. We denote by f_θ the representation map and by $\hat{Y}_\theta : \mathcal{X}_S \to [K]$ the clustering output (for any observed-view set S).

Loss family $\mathcal{F}_{rec}$. A reconstruction branch lies in $\mathcal{F}_{rec}$ if it admits a per-sample decomposition

$$\mathcal{L}(\theta; \mathcal{D}, \mathbf{M}) \;=\; \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{|O_i| \ge 2\} \cdot \tilde{\ell}\big(\theta;\, X^{(i)},\, M^{(i)}\big), \tag{5}$$

where $\tilde{\ell}$ is the per-sample reconstruction loss, defined only when at least two views of sample i are observed (i.e., the indicator $\mathbf{1}\{|O_i| \ge 2\}$ vanishes the term when $|O_i| < 2$). This captures cross-view reconstruction methods in which the reconstruction loss is computed over pairs of views co-observed for the same sample (§2).

Protocol probabilities. For a protocol P (a distribution over M), define

$$q_k(P) \;:=\; \mathbb{P}_{M\sim P}\big[|O| \ge k\big], \qquad k \in \{1, \dots, V\}. \tag{6}$$

For full-view reconstruction ($k = V$), $q_V(P) = p_c(P)$, the complete-sample proportion. Throughout, we work with $q_2(P)$ since $\mathcal{F}_{rec}$ requires only pairwise co-observation.

B.3.2 Assumption

Assumption B.1 (Bounded per-sample gradient)**.** *There exists $C_g < \infty$ such that for all θ in a compact neighborhood $\mathcal{N}(\theta_0)$ containing the optimization trajectory, and almost all (X, M) with $|O| \geq 2$, $\left\|\nabla_\theta \tilde{\ell}(\theta; X, M)\right\| \leq C_g$.*

The bound C_g is uniform along the optimization trajectory and does not grow as observation support decreases. Bounded loss values alone do not establish bounded parameter gradients. Uniform sampling from a fixed masked dataset gives the same argument with q_2 replaced by its empirical eligible fraction; sampling only eligible examples changes the normalization.

B.3.3 Lemma A: Gradient Magnitude Bound

Lemma B.1 (Gradient magnitude under protocol P)**.** *Under Assumption B.1, for any $\theta \in \mathcal{N}(\theta_0)$ and any sample $i \sim \mathrm{Unif}([n])$ with $M^{(i)} \sim P$ independent of past iterates,*

$$\mathbb{E}_{i,\, M^{(i)} \sim P}\left\|\nabla_\theta \ell_i(\theta)\right\| \;\leq\; C_g \cdot q_2(P). \tag{7}$$

Proof. By the definition of $\mathcal{F}_{rec}$ in (5), $\nabla_\theta \ell_i(\theta) = \mathbf{1}\{|O_i| \geq 2\} \cdot \nabla_\theta \tilde{\ell}(\theta; X^{(i)}, M^{(i)})$. Therefore

$$\begin{aligned}\left\|\nabla_\theta \ell_i(\theta)\right\| &= \mathbf{1}\{|O_i| \geq 2\} \cdot \left\|\nabla_\theta \tilde{\ell}(\theta; X^{(i)}, M^{(i)})\right\| \\ &\leq \mathbf{1}\{|O_i| \geq 2\} \cdot C_g \quad \text{(Assumption B.1)}.\end{aligned} \tag{8}$$

Taking expectation over i and $M^{(i)} \sim P$:

$$\mathbb{E}\left\|\nabla_\theta \ell_i(\theta)\right\| \leq C_g \cdot \mathbb{P}_{M^{(i)} \sim P}\big[|O_i| \geq 2\big] = C_g \cdot q_2(P). \qquad \square$$

B.3.4 Lemma B: Parameter Trajectory Bound

Lemma B.2 (Parameter displacement after T SGD steps)**.** *Let θ_t evolve under SGD with constant step size $\eta > 0$: $\theta_{t+1} = \theta_t - \eta\, \nabla_\theta \ell_{i_t}(\theta_t)$, where $\{i_t\}$ are iid uniform on $[n]$ and $\{M^{(i_t)}\} \sim P$ iid. Suppose all $\theta_t \in \mathcal{N}(\theta_0)$ for $t \in [0, T]$. Then*

$$\mathbb{E}\left\|\theta_T - \theta_0\right\| \;\leq\; \eta T \cdot C_g \cdot q_2(P). \tag{9}$$

Proof. By the triangle inequality,

$$\begin{aligned}\|\theta_T - \theta_0\| &= \left\|\sum_{t=0}^{T-1} (\theta_{t+1} - \theta_t)\right\| = \left\|\sum_{t=0}^{T-1} \eta \nabla_\theta \ell_{i_t}(\theta_t)\right\| \\ &\leq \eta \sum_{t=0}^{T-1} \left\|\nabla_\theta \ell_{i_t}(\theta_t)\right\|.\end{aligned} \tag{10}$$

Taking expectation and applying Lemma B.1 to each term (using that the iterate θ_t is independent of the future sample $(i_t, M^{(i_t)})$ in the standard SGD analysis):

$$\mathbb{E}\|\theta_T - \theta_0\| \leq \eta \sum_{t=0}^{T-1} \mathbb{E}\left\|\nabla_\theta \ell_{i_t}(\theta_t)\right\| \leq \eta T \cdot C_g \cdot q_2(P). \tag{11}$$

$\square$

Trajectory condition. Containment in $\mathcal{N}(\theta_0)$ is an assumption of the bound. A small expected displacement does not guarantee that every realized trajectory stays inside that neighborhood.

B.3.5 Proof of Theorem 4.2

Proof. Lemma B.2 gives the stated displacement bound. For a strict-complete loss, replace the support indicator in (5) by $\mathbf{1}\{|O_i| = V\}$ throughout Lemmas B.1–B.2. Its expectation is p_c, giving $\mathbb{E}\|\theta_T - \theta_0\| \leq \eta T C_g p_c$. $\square$

B.3.6 Scope

The argument applies to the gated branch with the normalization in (5). Other objectives can update shared parameters even when this branch is inactive. For an averaged mini-batch gradient, the triangle inequality preserves the same $C_g q_2$ upper bound; $1-(1-q_2)^B$ instead describes the probability of at least one eligible sample in an iid batch of size B. Adaptive optimizers require their own update analysis.

The theorem establishes an upper bound on parameter displacement. Accuracy also depends on the initial representation and decision margins, so small movement need not mean random clustering. Likewise, positive observation support does not establish convergence or an accuracy advantage over another protocol.

B.4 Cluster-Information Bound: Y is the Statistical Bridge

Corollary B.1 (Recovery does not increase available information at inference)**.** *Let $\hat{X}_v = g_\phi(\{X_u\}_{u\in O_i})$ be the output of a recovery module applied to the observed views at inference. Then $I(\{X_u\}_{u\in O_i},\, \hat{X}_v;\; Y) = I(\{X_u\}_{u\in O_i};\; Y)$.*

Proof. Direct from the data processing inequality: $\hat{X}_v$ is a deterministic function of $\{X_u\}_{u\in O_i}$, so appending it to the conditioning set cannot increase information about Y. □

This concerns inference-time information content. During training, recovery modules provide a useful gradient signal through cross-view reconstruction. The following inequality concerns distinct views.

Proposition B.1 (Cross-view predictive information)**.** *Under Assumption 4.1, for any fixed encoder f_θ and any two distinct views $u \neq v$:*

$$I(f_\theta(X_u);\; X_v) \;\leq\; I(f_\theta(X_u);\; Y). \tag{12}$$

The inequality is pointwise in the encoder and does not assert that both information quantities increase together during optimization.

Proof. Since $f_\theta(X_u)$ is a deterministic function of X_u, and $X_u \perp X_v \mid Y$ by Assumption 4.1, $f_\theta(X_u) \perp X_v \mid Y$, giving $I(f_\theta(X_u); X_v \mid Y) = 0$. Applying the chain rule:

$$\begin{aligned} I(f_\theta(X_u);\; X_v) &= I(f_\theta(X_u);\; X_v, Y) - I(f_\theta(X_u);\; Y \mid X_v) \\ &\leq I(f_\theta(X_u);\; X_v, Y) \\ &= I(f_\theta(X_u);\; Y) + \underbrace{I(f_\theta(X_u);\; X_v \mid Y)}_{=0}. \end{aligned}$$

□

Interpretation. Eq. (12) says that the cross-view predictive information is bounded above by the cluster-relevant information. Under conditional independence, Y is the sole statistical bridge between X_u and X_v: any feature of X_u that is predictive of X_v must pass through Y. The inequality concerns an encoder of X_u alone. CRAFT's fusion with all views available also observes the target view X_v, so this argument does not identify its reconstruction loss with cluster-information maximization.

B.5 Attention Reweighting and Exact Exclusion

Proposition B.2 (Attention Leakage Bound for Placeholder Tokens)**.** *Consider attention over V view tokens, with $k < V$ absent positions and retained-token logits bounded below by $s_{\min}$. When absent views are replaced by learnable placeholder tokens with logits $\leq \varepsilon$, their total attention weight satisfies*

$$\sum_{j\in\text{mask}} \alpha_j \;\leq\; \frac{k}{V-k} \cdot \exp(\varepsilon - s_{\min}).$$

Proof. The mask set has k elements with scaled dot product $\leq \varepsilon$: $\sum_{j\in\text{mask}} \exp(\mathbf{q}^\top \mathbf{k}_j/\sqrt{d}) \leq k \cdot \exp(\varepsilon)$. The denominator $Z \geq (V-k)\cdot\exp(s_{\min})$. Dividing yields the claim. □

Corollary (Exact exclusion under attention masking). Under canonical CRAFT, $\varepsilon = -\infty$ (attention masking sets the logit to $-\infty$ before the softmax) and the bound yields $\sum_{j\in\text{mask}} \alpha_j = 0$ exactly. This guarantees that absent views contribute zero information to the CLS representation, giving exact exclusion as required by C2. This forward property alone does not specify the accuracy response to removing an informative view.

Empirical verification. On HandWritten (V=6) with k=1, the learnable-placeholder variant allocates 9.4% of CLS attention to the placeholder versus the 16.7% uniform share, confirming that the trained model systematically aligns real-token keys more closely with the CLS query but does not achieve exact exclusion—only canonical attention masking does.

B.6 MFT as Weighted View-Subset Ensemble

Proposition B.3 (View Dropout Ensemble)**.** *In MFT, each sample draws* $k \sim \text{Uniform}(\{0, \ldots, V-2\})$ *views to exclude. The expected MFT loss equals a weighted sum over view subsets:*

$$\mathbb{E}_{\mathbf{m}}[\mathcal{L}(f_\theta(\mathbf{x}, \mathbf{m}))] = \sum_{S:|S|\geq 2} P(S) \cdot \mathcal{L}(f_\theta(\mathbf{x}_S)),$$

where $P(S) = 1/((V-1)\binom{V}{V-|S|})$.

Proof. Each mask determines a unique retained subset. The probability of a specific subset S with $|S| = V - k$ is $\frac{1}{V-1} \cdot \frac{1}{\binom{V}{k}}$. Normalization is verified by $\sum_{k=0}^{V-2} \binom{V}{k}/((V-1)\binom{V}{k}) = 1$. □

Interpretation. (1) The full-view subset (k=0) appears with probability $1/(V-1)$, retaining complete inputs in the training objective. (2) At V=6, 57 distinct subsets participate in the ensemble, providing strong regularization; at V=2, only 1 subset (full data) participates, so MFT performs no view deletion on CUB. (3) This identity expresses a weighted training objective; inference uses one forward pass for the observed subset, and the identity provides no accuracy or ensemble-improvement guarantee.

C Protocol Analysis

C.1 Closed-Form Protocol Formulas

For each protocol introduced in §3.2, we state the closed-form expressions for the realized entry-wise missing rate $\hat{r}$ and the complete-sample proportion p_c in the limit $n \to \infty$. Throughout, $r \in [0, 1]$ denotes the nominal missing rate as reported by each method.

Protocol 1 (sample-level; delete one view per sample). A fraction r of samples are designated incomplete; for each incomplete sample, exactly one view is deleted uniformly at random. Then:

$$\hat{r}_{\text{P1}} = \frac{r}{V}, \qquad p_c^{\text{P1}} = 1 - r.$$

Protocol 2 (protected view; entry-wise at r/V). For each sample, exactly one view is randomly designated as "protected" (always observed); the remaining $V-1$ views are masked by independent Bernoulli$(r/(V-1))$ trials, calibrated to realize the nominal entry-wise rate r/V in expectation. Then:

$$\hat{r}_{\text{P2}} = \frac{r}{V}, \qquad p_c^{\text{P2}} = \left(1 - \frac{r}{V-1}\right)^{V-1}.$$

Protocol 3 (entry-wise Bernoulli at r; one-hot guarantee). Each entry $M_v^{(i)}$ is masked by an independent Bernoulli(r) trial; any sample that loses all V views has one randomly chosen view restored (the one-hot guarantee). Then:

$$\hat{r}_{\text{P3}} = r - \frac{r^V}{V}, \qquad p_c^{\text{P3}} = (1 - r)^V.$$

Protocol 4 (protected view; fixed deletion budget at r). For each sample, one protected view is fixed. A total of $D = \min(\mathrm{round}(rnV), n(V-1))$ entries are deleted uniformly without replacement from the $N = n(V-1)$ non-protected entries. Here round rounds to the nearest integer, with ties to even. Thus $\hat{r} = D/(nV)$. The probability that a given sample stays complete is $\binom{N-(V-1)}{D}/\binom{N}{D}$, interpreted as zero when too few entries remain. For fixed V, its large-n limit gives

$$\hat{r}_{\mathrm{P4}} = \min\left(r, \frac{V-1}{V}\right), \qquad p_c^{\mathrm{P4}} = \left[\max\left(0, 1 - \frac{Vr}{V-1}\right)\right]^{V-1}.$$

The finite-sample deletion budget couples entries; independent Bernoulli trials give the same limit but a different mask law.

Numerical separation. At low r, the four formulas for p_c are nearly indistinguishable ($p_c \to 1$ as $r \to 0$). At moderate-to-high r and $V \geq 3$, they diverge by orders of magnitude—e.g., on V=6, r=0.7: $p_c^{\mathrm{P1}} = 30\%$ vs. $p_c^{\mathrm{P4}} \approx 1.0 \times 10^{-4}$, a $\sim$3,000$\times$ gap at the same nominal rate. Numerical instantiations appear in Table 7.

C.2 Monte Carlo Verification of Protocol Formulas

We verify the closed-form expressions for $\hat{r}$ and p_c through Monte Carlo simulation (n=2000, V=6, 100 independent trials per combination). Empirical means match analytical predictions within 0.5% in all cases.

Protected-view ceiling. The protected view imposes an upper bound of $(V-1)/V$ on $\hat{r}$ under Protocol 4. When V=2, r=0.5 and r=0.7 generate the same saturated mask distribution (and identical masks when the protected-view draw is shared). When V=3, the ceiling is $2/3 \approx 0.667$, so r=0.7 cannot be fully realized. Thus these two nominal rates are not distinct observation conditions for two-view datasets under Protocol 4.

DSIMVC protocol equivalence. DSIMVC uses per-sample Bernoulli trials rather than the fixed-count sampling of Protocol 1. By the law of large numbers, the fraction of incomplete samples converges to r as $n \to \infty$; for n=2000, the standard deviation is $\sqrt{r(1-r)/n} < 1.2\%$. The two procedures share these limiting statistics, while fixed-count and Bernoulli sampling have different finite-sample count variability.

C.3 Hidden-Variable Numerical Profiles

Table 7 instantiates the closed-form expressions for two representative configurations.

Table 7: Hidden variables ($\hat{r}$ and p_c) under four protocols, computed from the limiting formulas in §C.1.

	V=2, r=0.5		V=6, r=0.7	
Protocol	$\hat{r}$	p_c	$\hat{r}$	p_c
1	25.0%	50.0%	11.7%	30.0%
2	25.0%	50.0%	11.7%	47.0%
3	37.5%	25.0%	68.0%	0.073%
4	50.0%	0%	70.0%	0.010%

The right-hand block (V=6, r=0.7) exhibits a $\sim$4500$\times$ divergence in p_c between Protocol 2 (47%) and Protocol 4 (0.010%) at the same nominal missing rate—a difference in complete-sample support at the same nominal rate. Even the intermediate comparison Protocol 2 vs. Protocol 3 (47% vs. 0.073%) shows a $640\times$ divergence.

C.4 Protocol Mappings in Nine Implementations

To determine which protocol each method actually uses, we inspected the official code repositories of nine IMVC methods published between 2021 and 2025. Table 8 records the masking mechanism observed in each implementation and identifies the closest matching protocol from §C.1.

Table 8: Masking mechanism in nine IMVC implementations, mapped to the closest protocol from §C.1. ‡ shared masking code; † per-sample Bernoulli vs. fixed-count (§C.2); + Protocol 2 with a deterministic complete-sample floor.

Method	Venue	Masking Mechanism	Proto.
COMPLETER‡ [15]	CVPR'21	r/V entry-wise + protected view + calibration	2
DCP‡ [16]	TPAMI'22	r/V entry-wise + protected view (shared codebase)	2
DSIMVC	ICML'22	per-sample Bernoulli, post-fix $\lvert O_i\rvert \geq 1$	$1^{\dagger}$
ProImp‡ [13]	IJCAI'23	r/V entry-wise + protected view	2
APADC‡ [35]	TIP'23	r/V entry-wise + protected view	2
DVIMC [1]	MM'25	sample-level; delete $1{\sim}V{-}1$ views per incomplete sample	1
DCG‡ [39]	AAAI'25	r/V entry-wise + protected view	2
Energy-DIMC [29]	MM'25	μ/V entry-wise + per-sample protected view	2
HSACC [4]	NeurIPS'25	$(1{-}r)n$ complete samples + entry-wise masking on rest	2^{+}

The realized protocol distribution is concentrated: **seven of nine methods** sit at Protocol 2, and **the remaining two** (DSIMVC, DVIMC) use sample-level deletion. DVIMC deletes a variable number of views, so its mapping is a mechanism-level analogy, not an exact match to Protocol 1's effective-rate formula. The stricter regimes (Protocols 3 and 4) are not represented among these nine audited implementations, motivating their inclusion in the deployment and incomplete-training evaluations.

D Complete Experimental Results

D.1 CUB ($V{=}2$)

CRAFT uses one canonical checkpoint per seed ($\lambda_{\text{repr}}{=}0.1$, $\tau{=}0.5$, no MFT); baselines retrain per configuration. Under Protocol 4 with $V{=}2$ the protected view caps $\hat{r}$ at 0.5, so Protocol 4 $r{=}0.7$ has the same saturated mask law as $r{=}0.5$ (§C.2).

D.2 HandWritten ($V{=}6$)

Per-(protocol, rate) accuracy on HandWritten under canonical CRAFT (Stage 1 + Stage 2 + MFT, per §6.1); baselines retrain per configuration, while CRAFT starts from training data with all views available and reuses one final checkpoint per seed.

D.3 MultiFashion ($V{=}3$)

Table 11 reports per-(protocol, r) accuracy on MultiFashion; baselines retrain per configuration, while CRAFT starts from training data with all views available, with MFT enabled at $V{\geq}3$.

D.4 UCI-Digit, Out-Scene, and Caltech

A single CRAFT checkpoint per dataset and seed is evaluated across all 16 (protocol, r) configurations, mirroring the train-once protocol in §6.1; Table 12 reports the results. Baselines were not re-run on these three supplementary benchmarks.

The same pattern holds on all three benchmarks: lenient protocols (P1, P2) stay nearly flat as r grows, while stringent protocols (P3, P4) produce larger deployment-time degradation as fewer views are retained. Caltech has larger variation at high missing rates; the experiment does not isolate its cause from view heterogeneity or the training recipe.

D.5 YTF-31 ($V{=}5$, $K{=}31$)

Table 13 reports per-(protocol, r) accuracy on YTF-31; the train-once advantage compresses a 16-run DCP grid ($\sim$16 hours) into a single CRAFT run ($\sim$3 hours).

Table 9: ACC (%) on CUB (V=2, K=10). †Protocol 4 r=0.7 has the same mask law as r=0.5. *BURG cited from its paper; — = not reported. DCG has one effective seed; independent-run counts follow Appendix A.1.

Method	r	Proto. 1	Proto. 2	Proto. 3	Proto. 4
DCG	0.1	71.50	74.11	70.50	70.28
	0.3	72.56	70.33	67.78	65.39
	0.5	73.33	69.11	60.17	15.61
	0.7	63.50	59.16	55.95	17.72
COMPLETER	0.1	51.33	55.80	63.23	61.23
	0.3	61.60	62.43	52.10	18.27
	0.5	49.83	53.67	19.43	36.43
	0.7	18.90	18.50	21.13	36.43
DCP	0.1	65.00	59.83	62.33	64.17
	0.3	57.50	62.67	44.83	29.67
	0.5	51.33	51.83	28.17	36.83
	0.7	26.00	27.00	34.33	36.83
Energy-DIMC	0.1	75.20	68.55	71.48	71.29
	0.3	67.77	74.61	65.62	53.52
	0.5	58.20	65.23	47.85	42.38
	0.7	51.56	59.77	45.51	42.38
BURG*	0.1	—	71.50	—	—
	0.3	—	58.00	—	—
	0.5	—	57.33	—	—
	0.7	—	53.83	—	—
HSACC	0.1	63.13±1.2	63.63±2.2	62.23±2.4	59.50±2.3
	0.3	54.00±3.2	57.20±4.1	45.03±4.0	33.13±1.9
	0.5	43.40±3.2	45.63±3.1	30.70±0.6	38.63±1.5
	0.7	29.67±0.5	30.67±1.4	35.00±0.6	38.63±1.5
CRAFT	0.1	82.09±1.0	82.35±0.9	80.92±1.2	80.89±1.2
	0.3	79.19±1.1	79.44±1.3	76.27±1.6	74.15±2.2
	0.5	76.04±2.2	76.41±1.9	72.29±2.4	68.99±3.8
	0.7	73.61±2.7	73.23±2.7	69.81±2.9	68.99±3.8†

D.6 Missing-Rate-Matched Training

To compare training with all views available and rate-matched training on HandWritten, we train a variant exposed to the target missing rate during training, using hyperparameters appropriate for this training regime (§6.1).

Across the 8 configurations of Table 14, the two variants differ by at most 2.91pp; the per-rate variant occasionally outperforms the model trained with all views available at the most stringent configurations (Protocol 4, $r \geq 0.5$). The small gaps describe these tested recipes; training distribution and hyperparameters both vary, so this comparison does not isolate architecture as the sole cause.

D.7 Complete-MVC Comparison

CRAFT exceeds the listed complete-MVC baselines on CUB, UCI, HandWritten, and Caltech, while SparseMVC is higher on Out-Scene (Table 15). The same CRAFT model is also evaluated under missing views in the deployment grid.

D.8 Training Time Details

CRAFT starts from training data with all views available and reuses one final checkpoint per dataset and seed for every (protocol, r) configuration, while baselines retrain per configuration. On MultiFashion, the recorded DVIMC grid cost is $16 \times 8 = 128$ minutes, versus 14.5 minutes for one CRAFT training run; their ratio is $8.8\times$. A single DVIMC run is shorter than a single CRAFT run. On YTF-31 the recorded DCP grid and CRAFT run take about 16 and 3 hours, respectively. These

Table 10: ACC (%) on HandWritten (V=6, K=10). CRAFT and HSACC: 5-seed mean $\pm$ std. Energy-DIMC at $r = 0.7$ and DCP at $r = 0.7$ under Protocols 1/4 are single-seed. Other baseline entries without supplied std have no established independent-run count. †BURG cited from its paper; — = not reported.

Method	r	Proto. 1	Proto. 2	Proto. 3	Proto. 4
DCP	0.1	82.35	82.30	79.95	82.10
	0.3	83.90	84.00	22.45	24.60
	0.5	80.00	93.65	18.15	18.15
	0.7	71.55	70.10	16.40	16.25
Energy-DIMC	0.1	96.65	96.26	96.43	96.09
	0.3	96.37	96.54	95.20	94.87
	0.5	96.09	96.09	92.24	92.75
	0.7	96.37	95.65	83.98	83.26
BURG†	0.1	—	95.70	—	—
	0.3	—	95.30	—	—
	0.5	—	94.30	—	—
	0.7	—	79.75	—	—
HSACC	0.1	76.53±3.7	80.48±8.6	87.89±7.4	89.39±1.3
	0.3	87.91±4.2	85.40±4.1	14.92±0.2	15.05±0.2
	0.5	87.10±5.0	84.10±5.8	14.08±0.5	14.24±0.2
	0.7	80.56±1.7	83.48±1.3	13.80±0.1	13.52±0.6
CRAFT	0.1	97.60±0.8	97.61±0.8	97.49±0.9	97.53±0.8
	0.3	97.57±0.8	97.59±0.8	96.51±0.9	96.82±0.8
	0.5	97.57±0.8	97.53±0.8	92.92±1.3	93.69±1.1
	0.7	97.58±0.9	97.49±0.9	86.03±2.1	85.21±2.4

are historical training-time records; the retained timings do not itemize checkpoint selection, hyperparameter search, or the separately described post-Stage 2 MFT. They therefore do not establish a speedup for the complete development pipeline or the new per-configuration training comparison.

D.9 Extended Observation-Support Diagnostics

Observation-support comparison configurations. Table 16 reports three HandWritten configurations. Setting C uses Protocol 1 at $r = 1$, so every sample retains five views: $p_c = 0$, but $q_2 = 1$. Settings A and C have different effective missing rates (11.7% and 16.67%), so their contrast does not isolate p_c causally. DCP and DCG are trained under the incomplete configurations; CRAFT and CRAFT-Core reuse checkpoints whose training starts with all views available. The table is therefore a support diagnostic with an explicit training/deployment distinction.

$\mathcal{F}_{rec}$ sub-families and the COMPLETER multi-view extension. $\mathcal{F}_{rec}$ branches differ in whether reconstruction requires fully-complete samples ($|O_i| = V$) or only pairwise co-observation ($|O_i| \geq 2$). DCP and DCG belong to the former; their reconstruction support in Proposition 4.1 is p_c, so this branch is inactive on Setting C. Their low ACC is a separate empirical observation in Table 16. The multi-view extension of COMPLETER used in this work instead adopts pairwise-complete training (reconstruction and prediction losses are computed on any view pair (i, j) for which both views are observed in a given sample). Under Min-1 (every sample retains $V-1$ views), each sample contributes $\binom{V-1}{2}$ co-observed view pairs and the eligible fraction is exactly $q_2(P) = 1$; pairwise reconstruction retains support, and COMPLETER (5-seed) retains $68.76 \pm 5.34\%$ on Setting C. This result makes the distinction between complete-sample and pairwise support explicit within $\mathcal{F}_{rec}$.

Per-baseline failure modes (CUB Protocol 4, r=0.5). Table 17 matches each baseline's observed behavior at this configuration to the available diagnostic evidence.

Energy-DIMC four-protocol comparison (HandWritten, r=0.7). CRAFT leads at every protocol on HandWritten (Table 18), with protocol-dependent gaps: from 1.21 points at Protocol 1 to 1.95 points at Protocol 4. The gap actually widens between these endpoints. The comparison

Table 11: ACC (%) on MultiFashion (V=3, K=10). CRAFT uses attention masking with MFT; CRAFT and HSACC report 5-seed means. Other baseline run counts follow Appendix A.1; — = runtime failure.

Method	r	Proto. 1	Proto. 2	Proto. 3	Proto. 4
DCP	0.1	86.04	74.92	72.57	75.24
	0.3	72.75	75.21	66.18	67.21
	0.5	72.26	80.46	63.78	69.36
	0.7	68.95	78.73	33.21	25.11
Energy-DIMC	0.1	93.30	93.88	91.20	91.97
	0.3	91.64	91.27	87.75	88.53
	0.5	90.19	90.38	83.09	79.77
	0.7	89.90	89.61	72.60	29.14
HSACC	0.1	97.46±0.8	98.00±0.2	96.02±0.2	94.34±3.5
	0.3	96.40±0.3	96.29±0.1	85.17±3.8	82.50±2.5
	0.5	91.67±3.9	94.33±0.4	65.48±2.8	49.80±1.6
	0.7	83.36±3.0	88.03±3.6	17.62±1.3	21.61±0.4
DVIMC	0.1	83.86	86.51	79.95	86.65
	0.3	86.62	88.67	83.51	84.09
	0.5	88.82	85.81	80.67	80.43
	0.7	87.21	84.87	74.42	—
CRAFT	0.1	93.21	93.20	92.73	92.73
	0.3	92.82	92.70	90.66	90.83
	0.5	92.44	92.11	88.67	88.12
	0.7	91.97	91.38	86.64	85.41

Table 12: CRAFT ACC (%) on UCI-Digit (V=3, K=10), Out-Scene (V=4, K=8), and Caltech (V=6, K=20). 5-seed mean ± std.

Dataset	r	Proto. 1	Proto. 2	Proto. 3	Proto. 4
UCI-Digit	0.1	90.94±0.6	90.87±0.5	90.22±0.4	90.17±0.5
	0.3	90.59±0.4	90.08±0.3	85.75±1.8	85.61±1.9
	0.5	90.34±0.3	89.12±0.5	79.12±4.0	77.65±4.3
	0.7	90.14±0.6	87.64±1.1	72.93±5.9	68.31±6.9
Out-Scene	0.1	75.05±1.7	75.06±1.6	74.46±1.6	74.47±1.5
	0.3	74.61±1.5	74.69±1.6	71.34±1.7	71.49±1.6
	0.5	74.53±1.6	74.18±1.7	66.09±1.9	65.86±1.9
	0.7	74.07±1.8	73.60±1.5	59.78±2.1	57.05±2.1
Caltech	0.1	57.92±1.7	57.88±1.7	57.63±1.8	57.62±1.8
	0.3	57.78±1.8	57.77±1.7	56.17±2.1	56.23±2.0
	0.5	57.58±1.7	57.63±1.8	53.31±2.9	53.59±2.7
	0.7	57.40±1.7	57.35±1.8	48.12±4.5	47.74±4.5

records performance under reduced observation; it does not show that either method reaches an information-theoretic limit.

D.10 Full Comparison with BURG

Under Protocol 2—BURG's inferred protocol (§A.3.4)—CRAFT leads BURG at every tested rate on CUB and stays within $\sim$0.2% of its lenient ceiling on HandWritten across r, while BURG itself develops a $\sim$15% cliff between r=0.5 and r=0.7 on HandWritten (full per-rate numbers in Table 9 and Table 10).

D.11 Fusion Architecture Spectrum (Full Tables)

Table 19 reports the architectural variant comparison summarized in §6.3 across all four protocols on HandWritten (V=6). The variants share per-view encoders + Stage 1 + Stage 2 training without

Table 13: ACC (%) on YTF-31 (V=5, K=31). Random baseline: 3.2%. DCP trains 16 times ($\sim$16 hours); CRAFT trains once ($\sim$3 hours). Single seed (seed 42) due to dataset scale ($n \approx 100$k).

Method	r	Proto. 1	Proto. 2	Proto. 3	Proto. 4
DCP	0.1	32.97	33.29	27.95	33.85
	0.3	29.49	30.86	16.00	12.73
	0.5	28.05	33.04	10.80	11.17
	0.7	27.57	30.66	5.79	6.08
CRAFT	0.1	27.64	27.56	26.01	25.98
	0.3	26.88	26.76	22.39	22.27
	0.5	26.13	25.93	19.68	19.45
	0.7	25.39	25.16	18.45	18.10

Table 14: HandWritten (V=6): CRAFT trained with all views available vs. rate-matched CRAFT under Protocol 1 and Protocol 4. Both variants disable masked fine-tuning; their training distributions and hyperparameters differ. The canonical CRAFT* row in Table 2 uses a separate recipe with MFT. † = single-seed evaluation; all other entries are 5-seed mean.

	Protocol 1				Protocol 4			
r	0.1	0.3	0.5	0.7	0.1	0.3	0.5	0.7
All views available	97.37	97.50	97.69	97.83	97.48	95.68	90.56	82.48
Per-rate	94.75	94.59†	94.99	95.03	94.92	93.98	91.05	82.56
Gap	+2.62	+2.91	+2.70	+2.80	+2.56	+1.70	−0.49	−0.08

MFT for the within-table fusion comparison, differing only in the fusion block: (i) Transformer with attention masking (the CRAFT fusion block, strict C2); (ii) *SetMLP*, a DeepSets-style fusion [38] consisting of a per-view shared MLP, a masked mean pool over observed views, and a post-pool MLP of hidden dimension h' (strict C2), evaluated at two capacity tiers *SetMLP-Light* (h'=192) and *SetMLP-Match* (h'=560, fixed across datasets); and (iii) a zero-fill concat fusion baseline (soft C2). DVIMC uses a different training recipe and has recorded initialization failures (Appendix A.3.5). Canonical CRAFT in Table 10 uses a separate recipe with MFT. 5-seed mean ± std.

Three findings. *(1) Architecture dominance.* The tested Transformer dominates both SetMLP tiers and the fixed-input concat fusion baseline across every (protocol, rate) configuration, with the gap widening monotonically under stringent regimes; at the hardest tested condition the three strict-C2 architectures and the soft-C2 violator span $\sim$24 points. *(2) A larger SetMLP does not close the gap.* Increasing SetMLP's hidden size from h'=192 to h'=560 (parameters comparable to the concat fusion baseline) does not narrow the gap to the Transformer; the larger SetMLP-Match *underperforms* the smaller SetMLP-Light by $5-7\%$ under hard missingness. Thus increasing this hidden width alone does not improve these results; other capacity and optimization effects remain possible. *(3) The concat-fusion gap grows with r.* In the reported configurations, SetMLP-Light's lead over the fixed-input concat fusion baseline grows from a fraction of a point at r=0.1 to roughly 5–6% at r=0.7 under Protocol 4. This comparison changes the fusion block and cannot assign the full gap specifically to placeholder leakage.

The Transformer's query-dependent aggregation performs better than the tested shared-MLP pooling recipes. This supports the choice of fusion block empirically, without making it the unique architecture defined by C1+C2.

DVIMC implementation comparison. DVIMC's HandWritten grid includes initialization failures and non-monotone accuracy across successful runs; details appear in Appendix A.3.5. CRAFT completes every condition in its deployment grid. The implementations and training recipes differ, so this comparison does not prove C2 necessity.

Table 15: Clustering performance with all views observed (ACC %). † = numbers cited from original papers; OOM = out of memory on a single RTX 3090 (24 GB); — = not run.

Method	CUB	MF	UCI	OS	HW	Cal	YTF
EPFMVC†	—	—	—	—	96.90	—	—
SparseMVC [19]	77.07±1.1	OOM	90.27±3.5	77.49	90.26±3.5	47.10±1.1	OOM
CRAFT	83.80±1.1	93.41±0.3	91.46±0.5	75.55±0.8	97.49±0.9	58.86±1.2	28.10

Table 16: Observation-support comparison on HandWritten (V=6, K=10). ACC (%); Setting C DCP, DCG, and Energy-DIMC are 5-seed mean ± std. Setting A/D DCP and DCG and all Energy-DIMC A/D entries are single-seed. CRAFT A/D reuse the five-seed deployment means; CRAFT/Core C have no established independent-run count, and Core D is a separate reduced-recipe result. Bold = best per setting; ‡ $\leq 40\%$ on K=10; † on DCG = the A/D entries are single-seed; — = not run.

	Setting A Proto 1, r=0.7 $\hat{r}$=11.7% p_c=30%	Setting C Proto 1, r=1.0 $\hat{r}\approx$16.67% p_c=0	Setting D Proto 4, r=0.7 $\hat{r}$=70% $p_c\approx$0
Class 1: $\mathcal{F}_{rec}$			
DCP [16]	71.55	19.79±2.42‡	16.25‡
DCG† [39]	62.30	12.29±0.42‡	12.75‡
Class 2: cross-sample			
Energy-DIMC [29]	96.37	94.14±1.07	83.26
Class 3: per-sample (C1+C2)			
CRAFT-Core (Stage 1 only)	—	89.75	54.83
CRAFT (canonical)	**97.58**	**96.34**	**85.21**

E Ablation and Sensitivity Details

E.1 Full Ablation Tables

All ablation experiments use 3 seeds and report Protocol 4 ACC (mean±std) unless otherwise noted. Each column removes one component from the within-table "Full" reference. All rows use the v4 learnable-placeholder mask strategy to hold mask handling fixed across ablations, so absolute values are not directly comparable to the canonical CRAFT* attention-masking results in Table 2 (see Appendix E.4 for mask strategy).

Stage 1 pre-training is essential. Skipping Stage 1 entirely (NO PRETRAIN) drops accuracy by $23-34\%$ on HW and roughly 14% on CUB at the hardest tested rates (Table 20). Stage 1's role is to learn representations that pre-align with the Stage 2 cluster head; without it, Stage 2 must drive both representation learning and clustering simultaneously, and the evaluated clustering-only recipe performs worse under missing-view evaluation.

Reconstruction is the load-bearing Stage 1 signal. Replacing Stage 1 with consistency-loss-only training (REPR ONLY, λ_{recon}=0) collapses performance to near-random on HW (at most 16.01% on K=10, where random is 10%) and to $15-51\%$ on CUB. The reconstruction loss provides the bulk of the cluster-relevant gradient; the consistency loss alone, despite being per-sample and C1-compliant, cannot anchor the representation in the absence of a per-view reconstruction target in this evaluated recipe.

Consistency loss has dataset-dependent effects. Removing the consistency loss (RECON ONLY, λ_{repr}=0) leaves accuracy approximately unchanged on HW ($\leq$ 1% gap to Full at every r) but *improves* accuracy on CUB by $3-8\%$ (Table 20). View count and dataset identity vary together here, so the asymmetry cannot be attributed to view count alone. Consistency is therefore a dataset-dependent regularizer, not a load-bearing component.

Table 17: CUB, Protocol 4, r=0.5 (p_c=0). Observed ACC and implementation diagnostics; these entries do not identify a common failure cause. Bold = best ACC.

Method	ACC (%)	Diagnostic context
DCG	15.61	Recovery gradient vanishes (Prop. 4.1)
ProImp	crash	Fixed-topology crash
COMPLETER	36.43	Partial recovery dependence
DCP	36.83	Partial recovery dependence
HSACC	38.63±1.5	Partial recovery dependence
Energy-DIMC	42.38	Distributional objective; low ACC
CRAFT	**68.99**±3.8	Per-sample independence + variable-length fusion

Table 18: HandWritten (V=6), r=0.7. Energy-DIMC retrains per rate (single seed). CRAFT: one canonical attention_mask checkpoint with MFT, 5-seed mean. Bold = best per protocol.

Method	Proto. 1	Proto. 2	Proto. 3	Proto. 4
Energy-DIMC	96.37	95.65	83.98	83.26
CRAFT	**97.58**±0.9	**97.49**±0.9	**86.03**±2.1	**85.21**±2.4

Stage 2 components contribute marginally. Removing the Stage 2 fine-tuning (NO STAGE 2) or its entropy regularizer (NO ENTROPY) reduces accuracy by at most $\sim$3.5% at every condition tested (Table 20). Both contribute to the tested recipe, with larger effects on CUB than on HandWritten.

Additional MFT has dataset-dependent effects. Masked fine-tuning (§5.3) yields modest single-digit improvements on HW and MultiFashion at high missing rates and a small degradation on CUB. The asymmetry follows the view-subset ensemble interpretation of MFT (Appendix B.6): at $V \geq 3$, MFT mixes many distinct view subsets and acts as a regularizer; at $V = 2$, the stated sampling rule retains all views. The observed changes therefore include further fine-tuning and any feature augmentation, rather than a view-deletion effect.

E.2 Reduced Core (CRAFT-Core)

CRAFT-Core retains attention masking and a Stage 1 reconstruction loss, removing SimSiam consistency, entropy and KL regularizers, MFT, and the Stage 2 cluster-head fine-tuning. A single Core checkpoint per dataset and seed is evaluated under all four protocols at four rates with attention masking applied at inference; Table 21 reports the result. Core retains useful accuracy on both datasets and exceeds the listed reconstruction baselines on CUB. On HandWritten it does not win every configuration. At Protocol 4, $r = 0.7$, Core reaches 51.94% versus canonical CRAFT's 85.21%. This comparison removes several components jointly, so their individual effects cannot be separated from the gap. The 54.83% reduced-recipe entry in the support diagnostic belongs to the separate original ablation recipe and is not this Core checkpoint.

E.3 Sensitivity Figures

Figure 4 reports the four sweeps at the canonical evaluation point (Protocol 4, r=0.5). We re-ran each sweep at $r \in \{0.1, 0.3, 0.7\}$ and confirmed that the optimal settings remain stable: L=1 is optimal at every tested missing rate; d=128 is optimal for CUB at every rate (larger d overfits on n=600); the λ_{repr} dataset-dependence (HW optimum near 0.5, CUB optimum near 0.1) is preserved across rates; the temperature sweep over $\tau \in [0.1, 0.7]$ has an ACC spread below 0.6 percentage points. The relative ordering of configurations in each panel does not change with r, so the canonical configuration in Appendix A.2 is appropriate at all four nominal rates.

E.4 Missing-View Strategy Comparison

The complete comparison is reported in Table 4 in §6.3.

In Table 4, the strict-C2 strategy (attention masking) consistently dominates soft-C2 alternatives (learnable, zero, mean), with the gap widening at higher missing rates. This is consistent with Proposition B.2: under attention masking, absent views contribute zero attention weight exactly;

Table 19: Fusion architecture spectrum on HandWritten (V=6): ACC (%) under all four protocols. Bold = best per configuration.

Architecture	r	Proto. 1	Proto. 2	Proto. 3	Proto. 4
Transformer + attn mask (no MFT)	0.1	**97.28**±1.0	**97.28**±1.0	**97.11**±1.0	**97.14**±1.1
	0.3	**97.25**±1.1	**97.26**±1.1	**95.31**±1.6	**95.59**±1.5
	0.5	**97.24**±1.1	**97.13**±1.1	**90.02**±3.0	**90.79**±3.0
	0.7	**97.24**±1.0	**97.09**±1.1	**80.29**±5.1	**79.06**±5.7
SetMLP-Light (h'=192)	0.1	94.80±1.3	94.78±1.3	93.83±1.2	94.01±1.2
	0.3	94.54±1.2	94.54±1.3	88.23±1.3	88.82±1.3
	0.5	94.37±1.2	93.94±1.3	78.19±1.6	78.97±1.8
	0.7	94.17±1.2	93.56±1.3	64.38±2.0	62.09±2.3
SetMLP-Match (h'=560)	0.1	94.56±1.6	94.53±1.6	92.84±1.5	92.93±1.7
	0.3	93.86±1.6	93.92±1.6	85.30±1.7	85.89±1.7
	0.5	93.45±1.6	93.11±1.6	73.23±1.6	73.83±1.7
	0.7	92.74±1.6	92.19±1.7	57.78±1.9	55.15±1.9
Concat fusion (fixed input, soft C2)	0.1	94.58±0.3	94.53±0.3	93.31±0.3	93.38±0.4
	0.3	94.17±0.2	94.27±0.3	86.13±0.4	86.89±0.2
	0.5	93.98±0.3	93.60±0.4	74.09±0.4	75.04±0.4
	0.7	93.63±0.3	92.93±0.3	58.72±0.6	56.55±0.7

Table 20: Loss and training-stage ablation (Protocol 4 ACC %). Bold = ablation column outperforming Full.

	r	Full	No Stage 2	No Entropy	Recon Only	Repr Only	No Pretrain
HW	0.1	94.23±1.1	94.20±1.0	94.12±1.0	93.66±1.5	16.01±2.4	62.45±3.8
	0.3	87.63±0.9	87.09±0.9	86.96±0.9	86.90±3.8	12.56±0.4	53.56±5.3
	0.5	74.42±1.3	73.62±1.5	73.47±1.6	74.22±5.8	11.33±0.1	43.15±6.4
	0.7	54.27±1.9	53.67±1.9	53.20±1.7	**54.83**±6.8	10.46±0.1	31.40±4.4
CUB	0.1	76.86±3.7	73.84±1.7	74.26±1.8	**79.99**±4.8	50.74±3.2	51.80±1.6
	0.3	68.62±4.4	65.63±1.9	66.03±2.1	**73.93**±4.8	31.60±1.8	49.34±1.3
	0.5	61.54±5.7	57.99±2.5	58.42±2.7	**69.28**±5.5	15.34±1.1	47.49±3.3

under soft-C2, the bound allows nonzero attention to missing positions. The ACC differences in this table remain empirical observations.

E.5 Aggregation Strategy

In Table 22, the three aggregation strategies (CLS / Mean / Max) differ by at most 4.28 percentage points, indicating that the C1+C2 properties are carried by the attention mechanism itself, not by the choice of pooling head.

E.6 Attention Analysis

On CUB (V=2), with one view missing the learnable placeholder receives approximately 38% of total CLS attention (76% of the uniform share), higher than the 57% ratio observed on HandWritten (V=6) because the shorter sequence leaves fewer real tokens to redistribute weight to. Under canonical attention masking, the missing position receives exactly zero weight regardless of the number of views (Proposition B.2, $\varepsilon = -\infty$); Figure 5 visualizes the placeholder attention reallocation pattern on HandWritten as a function of the number of missing views.

Table 21: CRAFT-Core ACC (%) on CUB and HandWritten under all four protocols. 3-seed mean ± std.

Dataset	r	Proto. 1	Proto. 2	Proto. 3	Proto. 4
CUB	0.1	79.96±3.5	79.79±3.4	78.36±2.9	79.01±3.6
	0.3	76.93±3.0	77.40±3.6	73.58±2.6	72.34±3.0
	0.5	74.30±2.9	74.33±3.0	69.98±2.6	66.93±3.2
	0.7	71.47±2.4	70.91±2.2	67.19±2.1	66.93±3.2
HW	0.1	93.63±0.9	93.60±1.0	92.39±1.6	92.29±1.7
	0.3	93.35±1.1	93.27±1.2	85.08±2.8	85.46±3.4
	0.5	92.92±1.5	92.68±1.6	71.90±3.8	72.34±3.8
	0.7	92.81±1.8	92.03±1.7	54.50±3.3	51.94±3.2

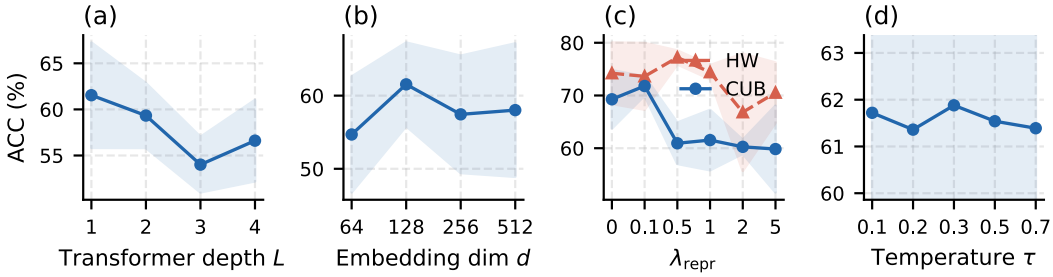


Figure 4: Sensitivity to four key hyperparameters under Protocol 4 r=0.5. **(a)** Depth: L=1 optimal; deeper fusion amplifies overfitting. **(b)** Embedding dim: d=128 optimal for CUB (n=600); larger d overfits. **(c)** λ_{repr}: dataset-dependent (HW optimum at 0.5, CUB at 0.1, consistent with the asymmetry in Table 20). **(d)** Temperature: insensitive (< 0.6 percentage-point spread over $\tau \in [0.1, 0.7]$). Shaded bands: ± 1 std over 3 seeds.

F Extended Discussion

F.1 Implications and Practical Recommendations

Cross-view reconstruction provides useful supervision when its required views are observed together. Our analysis bounds the gradient contribution of this branch under explicit support and normalization conditions. Three practical recommendations follow: report $\hat{r}$ and p_c alongside nominal rate r, and the relevant pairwise support where needed; check which loss branches remain active on the training masks; and assess incomplete training separately from reusing a model whose training starts with all views available. C1+C2 guide forward computation across observed subsets, while successful learning depends on the objective and the available information.

F.2 Reproducibility Statement

Code, configurations, and a four-protocol evaluation toolkit are released with the paper; CRAFT inference under arbitrary missing-view patterns uses only PyTorch's standard `MultiheadAttention` key-padding-mask interface, with no custom module required.

Table 22: Aggregation strategy comparison (Protocol 4 ACC %). Bold = best per row.

Dataset	r	CLS	Mean	Max
HW	0.1	**94.23**±1.1	93.85±0.1	93.45±0.6
	0.3	87.63±0.9	**88.72**±0.5	84.85±2.1
	0.5	74.42±1.3	**75.61**±1.5	71.33±3.7
	0.7	**54.27**±1.9	53.62±2.7	51.69±4.0
CUB	0.1	**76.86**±3.7	75.50±4.0	76.79±3.0
	0.3	68.62±4.4	68.10±3.7	**68.98**±3.0
	0.5	61.54±5.7	**61.97**±4.2	61.92±2.8

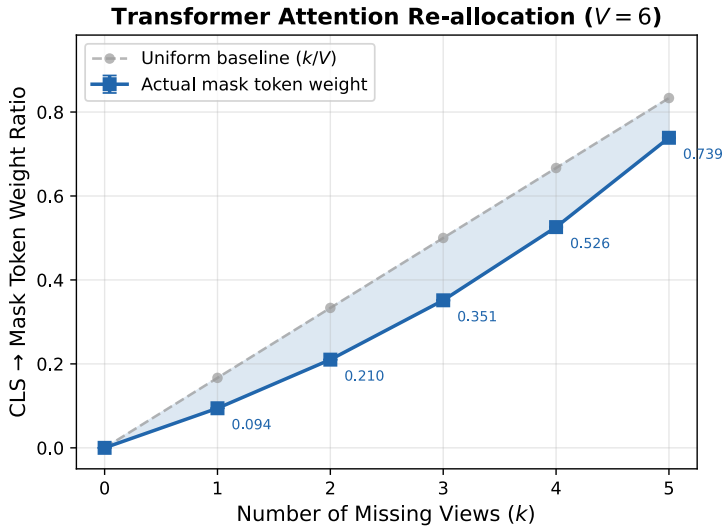


Figure 5: CLS→placeholder attention weight on HandWritten (V=6, learnable-placeholder strategy) vs. the number of missing views k. Trained model (solid) vs. uniform baseline k/V (dashed).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 describe protocol divergence, support-gated reconstruction bounds, and the CRAFT architecture. The experiments distinguish information-matched training comparisons from train-once deployment across seven benchmarks, starting from training data with all views available.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 7 identifies the synthetic missingness setting and the untested naturally occurring, quality-dependent, and correlated sensor failures. Training with an entirely unobserved modality or strictly zero complete samples beyond the reported conditions remains future work. Section 4 limits the reconstruction bounds to their stated loss family and assumptions.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Theorem 4.1, Proposition 4.1, and Theorem 4.2 state the applicable information, normalization, and optimization assumptions in Section 4; full statements and proofs appear in Appendix B. The separate role of the forward-architecture conditions C1 and C2 is discussed in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [No]

   Justification: The paper provides the architecture, training algorithm, deployment hyperparameters, seed protocol, and hardware. However, Appendix A.2 notes that the retained records do not establish the checkpoint-selection criterion for every historical run; the disclosure is therefore not complete.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Public repositories provide the CRAFT implementation and reproduction scripts (`https://github.com/dk23lhl/CRAFT`) and the standalone `imvc-audit` evaluation toolkit (`https://github.com/dk23lhl/imvc-audit`). All seven benchmarks (CUB, MultiFashion, HandWritten, UCI-Digit, Out-Scene, Caltech, YTF-31) are publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [No]

   Justification: Section 6.1 and Appendix A.2 describe the comparison conditions, deployment hyperparameters, optimizer, and training schedule. The paper does not give a complete per-condition configuration and selection record for every historical run.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Table 3 reports five-seed mean $\pm$ standard deviation, and Table 4 reports three-seed mean $\pm$ standard deviation. Appendix A.1 specifies the aggregation rules and single-seed exceptions, including YTF-31 and selected baseline entries. Standard deviations are shown only where available; these comparisons do not establish statistical significance.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Appendix A.1 specifies the hardware (single NVIDIA RTX 3090, 24GB VRAM, PyTorch 2.1, CUDA 12.1), and Appendix A.2 gives the training schedules. Historical timings in Appendix D.8 do not cover every experiment or itemize checkpoint selection, hyperparameter search, and post-Stage 2 MFT.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: This research adheres to the NeurIPS Code of Ethics. The experiments use existing benchmark datasets and do not recruit human participants or collect new participant data.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: Multi-view clustering is foundational unsupervised learning research with no easily predictable direct path to negative societal impacts.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: This work does not release high-risk models or datasets requiring safeguards; CRAFT is a clustering model trained on publicly available benchmarks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The datasets and baseline methods are credited, and the public CRAFT repository describes the expected data files and reproduction procedure. The manuscript does not enumerate the licenses and terms of use for every dataset and baseline implementation.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The public repositories document (1) the CRAFT implementation and reproduction scripts, and (2) the `imvc-audit` standalone evaluation toolkit (Python package with closed-form $\hat{r}$ and p_c formulas, mask sampling utilities, and a CLI) for cross-protocol IMVC audit.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This research does not involve human subjects, so IRB approval is not required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not a component of the core method.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.