University of Nottingham
UK | CHINA | MALAYSIA

Interim Report:
Investigation of Defence Mechanisms against Model Inversion Attacks

Submitted **December, 2025**, in partial fulfillment of
the conditions for the award of the degree **BSc Computer Science.**

Yixuan ZHANG

20513731

hnyyz39@nottingham.edu.cn

Supervised by Dr. Jianfeng REN

BSc (Hons) Computer Science

School of Computer Science
University of Nottingham Ningbo China

Abstract

Collaborative (edge–cloud) inference splits a network into an on-device encoder and a cloud decoder; the uploaded intermediate features are vulnerable to model inversion attacks (MIAs). Xia et al. [52] formalized a defence by maximizing a Gaussian-mixture lower bound of the conditional entropy $\mathcal{H}(x|z)$, but fitting high-dimensional mixtures is computationally heavy and sensitive to non-Gaussian feature geometry. I propose methods to target a learnable, distribution-agnostic surrogate of $\mathcal{H}(x|z)$ that preserves task accuracy while increasing inversion error in this Final Year Project. To date: (1) the public CEM baseline has been reproduced under the default CIFAR-10 split (VGG11-BN-SGM, cutlayer=4, noise variance 0.025, $\lambda = 16$); (2) a gated-attention conditional-entropy surrogate was designed, using class-wise gated pooling and variance-based log-entropy penalty, raising attack MSE from 0.0436 to 0.0473 and lowering SSIM from 0.432 to 0.411 with only a minor accuracy drop (Prec@1 85.18% to 84.34%); (3) an exploratory Slot + Gated Cross-Attention surrogate (slot aggregation plus Flamingo-style gated cross-attention) reached 84.69% accuracy but degraded privacy (MSE 0.0393, SSIM 0.459). The current experimental results indicate gated attention as a stronger privacy–utility trade-off. The slot-based route likely needs architectural fusion (e.g., shortcut or parallel coupling with gated pooling) to stabilize variance estimates before it can surpass the baseline. Next steps focus on such fusion designs and hyperparameter sweeps to retain accuracy while further elevating inversion error.

Contents

1 Introduction

1.1 Background and Motivation

Modern vision and multimodal networks are routinely deployed as services, yet their use in privacy-sensitive domains (health, biometrics, personal media) is constrained by the risk that intermediate computations reveal private inputs. Collaborative (edge–cloud) inference mitigates on-device compute and bandwidth by splitting the model: a shallow encoder runs on the device, producing an intermediate representation z that a cloud decoder consumes [40, 44, 46]. This architectural split leaves z as the primary attack surface. Model inversion attacks (MIAs) exploit z and any accessible parameters to reconstruct input content: early confidence-based attacks invert logits or posterior scores [13]; generative adversarial attacks synthesize inputs consistent with z [19]; gradient leakage recovers training data in white-box settings [54]; recent work shows that even large generative models can be forced to emit training samples [4]; and systematic white-box analyses demonstrate both passive and active inference threats against split and federated training [37]. These results show that the intermediate features sent between edge and cloud contain too much redundant information. If we don't constrain them, attackers can easily exploit this redundancy to reconstruct the input.

Conditional Entropy Maximization (CEM) [52] was proposed to give a effective solution on this redundancy. The pipeline mirrors collaborative inference: the local encoder F_e produces $z = F_e(x)$; isotropic Gaussian noise with covariance Σ_p is added to form $\tilde{z}$; the cloud decoder F_d computes the task loss L_D; and, in parallel, a k-component Gaussian Mixture Model is fitted to $\tilde{z}$ to approximate the conditional entropy of x given z. The surrogate

$$L_C = \sum_{i=1}^{k} \pi_i \Big(-\log \pi_i + \tfrac{1}{2} \log \tfrac{|\Sigma_i + \Sigma_p|}{|\Sigma_p|} \Big)$$

is added to the objective $L = L_D + \lambda L_C$, so that gradients from L_C push F_e to increase $\mathcal{H}(x|z)$ while L_D preserves utility. The architecture, cut position, and noise model remain unchanged; only the conditional-entropy term shapes privacy.

Defences for MIAs outside CEM fall into two families. Cryptographic protection (secure computation, homomorphic encryption) can in principle hide z, but current implementations impose prohibitive latency for high-throughput inference and are difficult to deploy on resource-constrained devices. Obfuscation-based defences reshape z to reduce leakage: noise injection and adversarial representation learning [25], pruning and Lipschitz-style regularization [10], distance-correlation minimization [47], and stochastic dropout [42]. These methods are largely heuristic and lack a direct link to a worst-case inversion bound, motivating a learnable, distribution-agnostic surrogate that retains CEM's theoretical spirit but avoids per-epoch mixture fitting.

The choice of cut position makes this even trickier. Shallow cuts save computation on the client but create high-dimensional and less task-specific features z that are easier to invert. On the other hand, deeper cuts are better for privacy but usually cost too much for edge devices. Because of this, the surrogate for $\mathcal{H}(x|z)$ needs to handle shallow, noisy features well, without requiring computationally expensive density estimation. Attention mechanisms compute data-dependent weights and moments, suggesting a path to replace mixture fitting with learnable, differentiable statistics that reflect dispersion without explicit clustering.

1.2 Problem Statement

The published CEM surrogate hinges on per-epoch GMM fitting, which becomes numerically fragile on shallow, noisy, high-dimensional smashed data, and gradients either vanish or explode, while the per-epoch clustering overhead is not suited to edge-side computational capability [8], such as smartphones with lightweight chips and low-power circuit design [9]. Mixture assumptions also struggle to capture heavy-tailed or multi-modal feature geometry produced by shallow cuts, making L_C unstable precisely where privacy pressure is most needed [36]. Under the fixed split architecture and strong white-box MIA threat model (encoder known, $\tilde{z}$ observed, reconstructor trained to convergence), the outstanding issue is that conditional-entropy regularization depends on a brittle density fit. The question is whether a drop-in surrogate can retain the information-theoretic intent while remaining stable, fully differentiable, and lightweight on non-Gaussian features.

1.3 Aim and Objectives

In this project the estimation of L_C in CEM architecture is changed while preserving the split architecture, noise model, and task loss. The first attempt was trying to implement a Slot Attention with Gated Cross Attention surrogate: slot attention [32] serves as learnable mixture components, and Flamingo-style gated cross-attention [1] refines token–slot alignment to estimate variance-based entropy signals. This design proved complex to tune and, under default CIFAR-10 settings, did not surpass the baseline—accuracy remained similar, but inversion robustness weakened. Guided to simplify, the subsequent gated-attention surrogate directly computes class-wise gated pooling and log-variance penalties; this yields higher inversion MSE with only minor accuracy loss. The next step is to fuse the two ideas—shortcut or parallel coupling of slots with gated pooling—to stabilize variance estimation while retaining the expressive capacity of slots.

2 Related Work

2.1 Model Inversion Attacks

Early work inverted fixed feature encoders, showing that even deep convolutional representations preserve recognizable pixels [12, 35]. Fredrikson et al. demonstrated that class posteriors alone can leak sensitive attributes [13], establishing that semantic signals in outputs are exploitable. Generative approaches strengthened MIAs: Hitaj et al. used GANs to synthesize inputs consistent with leaked activations in collaborative training [19], while Carlini et al. extracted training samples from diffusion models [4], indicating that powerful priors make inversion easier. Gradient-leakage attacks such as DLG recover raw training data from shared updates [54], and comprehensive white-box analyses [37] document both passive and active inference in centralized and federated regimes.

In the edge–cloud setting, He et al. [17] formalized attacks when the attacker knows the encoder and sees the smashed data z, showing that shallow cuts and high-dimensional features make inversion particularly effective. Follow-up work on edge–cloud systems confirmed that even modest architectural changes do not prevent leakage unless z is explicitly regularized [18]. These findings motivate defences that reduce redundant or task-irrelevant information in z while preserving downstream accuracy.

MIAs can be categorized by attacker knowledge and reconstruction mechanism. Confidence based and posterior attacks exploit soft outputs [13]; latent inversion uses a learned decoder given z (as in [17]); gradient-leakage reconstructs from parameter updates [54]; and generative priors (GANs or diffusion) hallucinate samples consistent with z [4, 19]. Across these categories, success correlates with the mutual information between x and z, reinforcing conditional entropy as a meaningful robustness lens.

2.2 Split Learning / Edge-Cloud Privacy

Split/edge-cloud training partitions networks to balance device efficiency and accuracy [40, 44, 46]. Cutting at early layers minimizes on-device compute but yields high dimensional, less task-specific z, which enlarges the attack surface [17]. Deeper cuts improve privacy by embedding more task-specific features, but exceed typical edge budgets. This tension has motivated a spectrum of defences that keep the split architecture intact:

- **Noise and adversarial representation learning.** Adding Gaussian noise or adversarially training encoders to resist a proxy decoder can reduce information leakage [25]. GAN-based defences further co-train an inversion adversary to harden z [14], though success depends on the proxy's strength.
- **Correlation minimization.** Distance-correlation penalties reduce statistical dependence between x and z [47], aiming to strip task-irrelevant redundancy while keeping discriminative cues.
- **Pruning and Lipschitz constraints.** PATROL [10] prunes channels under Lipschitz-style constraints to suppress leak-prone activations. Such structure-aware pruning improves robustness but can hurt accuracy if over-applied.
- **Stochastic masking.** Dropout [42] reduces consistency across queries, lowering memorization of fine-grained details in z, but its effect is dataset- and cut-dependent.
- **Robustness transfer.** ResSFL [28] transfers robustness across clients in split federated learning, showing that defensive signals can propagate even under heterogeneous data.

These methods demonstrate empirical gains yet remain heuristic: they assume specific proxy attackers, require hyper-parameter sweeps, and lack formal links to worst-case inversion bounds. CEM [52] differs by introducing an information-theoretic objective that, in principle, regularizes z irrespective of the attacker's architecture, but its practical utility hinges on the stability of the entropy surrogate.

2.3 Conditional Entropy Maximization and Surrogates

Conditional entropy links directly to inversion difficulty: higher $\mathcal{H}(x|z)$ raises a lower bound on reconstruction MSE under the assumed noise model [52]. CEM instantiates this by adding $L = L_D + \lambda L_C$, where L_C is a differentiable log-determinant surrogate of $\mathcal{H}(x|z)$ obtained from a Gaussian mixture fitted on noisy features [52]. This design is attacker-agnostic in principle, but its practicality is limited by: (i) GMM fragility on shallow, non-Gaussian, heavy-tailed features; (ii) sensitivity of mixture assumptions to multi-modality; and (iii) per-epoch density estimation overhead [8, 18]. Alternative regularizers (distance correlation [47], adversarial noise [25], pruning [10], dropout [42])

reduce leakage but do not optimize an explicit entropy bound. These gaps motivate a distribution-agnostic surrogate that captures dispersion without clustering, remains stable under back-propagation, and can be injected into the same split-inference protocol.

2.4 Attention and Self-Attention Foundations

Attention assigns data-dependent weights to tokens and underpins modern sequence and vision models. Early encoder–decoder alignment for NMT [2, 34] evolved into fully self-attentive architectures such as Transformers [45] and Vision Transformers [11], where pairwise affinities within a set enable dynamic aggregation without fixed receptive fields. Scaling variants address efficiency and inductive bias: non-local blocks bring self-attention to CNNs for long-range context [49]; sparse/low-rank forms (Linformer [48], Performer [5], Longformer [3], BigBird [53]) reduce quadratic costs; hierarchical designs (Swin [31], ConViT [6]) inject locality for stability on images; and set-focused models (Set Transformer [27]) provide permutation-invariant aggregation. These mechanisms offer a toolbox for computing differentiable, data-dependent statistics over feature sets—precisely what is needed to replace static mixture fitting in entropy surrogates.

2.5 Gated Attention Mechanisms

Gated variants moderate attention or feature flow with learnable gates to improve stability and control variance. Channel and spatial gates in SENet [20] and CBAM [51] re-weight activations; non-local blocks with gated residuals stabilize long-range interactions [49]; GLUs gate convolutional features [7]; MIL-style gated pooling stabilizes instance weighting [23]; and many hybrid CNN/Transformer models add gating to prevent overconfident token mixing [11]. These designs share a theme—learnable modulation to avoid overconfident assignments and to smooth gradients.

For a CEM surrogate, gated pooling over class-specific features yields weighted means and variances as differentiable dispersion statistics, avoiding iterative clustering and reducing sensitivity to initialization. Gates can be applied per channel, per token, or jointly, adapting to the structure of z (e.g., channel attention for convolutional maps, token attention for flattened features). Because gates introduce few parameters and are trained end-to-end, they back-propagate through noisy, high-dimensional z more gracefully than EM-style GMM fitting. Beyond sequence tasks, gated attention has improved stability in multimodal and dense prediction models [49, 51], suggesting resilience to the noisy, shallow features produced in split inference when Gaussian noise is injected as in CEM.

2.6 Slot Attention

Slot attention [32] learns a small set of latent "slots" that act like learnable mixture components, assigning tokens through iterative attention and GRU updates; it has been extended to object-centric learning and structured scene parsing. Related set-based architectures (Set Transformer [27]) and hybrid convolution/attention designs (ConViT [6]) also aim to aggregate tokens into compact latent sets. In the context of conditional-entropy surrogates, slots can serve as adaptive mixture components without explicit density estimation, potentially capturing multi-modality in z. However, slot-based models introduce sensitivity to hyper-parameters (number of slots, temperature, update depth) and can be prone to collapse without additional regularization—an important consideration when z is shallow, noisy, and high-dimensional. Stabilizing slots often requires

careful gating, norm layers, or auxiliary losses, increasing tuning burden in split-learning regimes.

2.7 Cross-Attention and Gated Cross-Attention

Cross-attention aligns queries with key–value memories and is widely used in multimodal and hierarchical models: ViLBERT [33] and LXMERT [43] for vision–language pre-training, ALBEF [30] and BLIP-2 [29] for efficient alignment, Flamingo [1] for few-shot VLMs, and Perceiver [24] for latent bottlenecks. Gated residuals are often added to stabilize training and prevent over-smoothing. When combined with slots, cross-attention sharpens responsibilities of tokens to latent components, acting as a learnable analogue to mixture assignment; gating modulates binding strength to prevent collapse.

3 Methodology

3.1 Baseline Overview

The methodological starting point for this study is the Conditional Entropy Maximization (CEM) framework proposed by Xia et al. [52]. This work marked a major change in how defences are designed, moving away from heuristics methods toward approaches based on solid information-theoretic proofs. The authors formally proved that the conditional entropy $\mathcal{H}(x|z)$ of the input x given the intermediate feature z constitutes a theoretical lower bound on the reconstruction Mean Square Error (MSE) against any worst-case adversary. To calculate this hard-to-compute value, the baseline employs a variational approximation strategy: it models the latent feature distribution using a Gaussian Mixture Model (GMM) and derives a differentiable surrogate loss, L_C, which penalizes the mutual information between the input and the smashed data. By integrating this surrogate into the training loop, the encoder is incentivized to maximize feature dispersion, thereby raising the barrier for inversion attacks.

3.2 Proposed Method 1: Gated-Attention CEM

Method 1 pursues the most conservative change that still captures intra-class dispersion. Drawing on gated pooling in vision [20, 51] and MIL [23], each class slice of $\tilde{z}$ is softly re-weighted by a learnable gate. Gated means and variances are then computed in closed form, and a hinge on log-variance, calibrated to the injected noise, becomes the surrogate L_C^{gated}. The design keeps three invariants: (i) differentiability everywhere (no hard assignments), (ii) robustness to outliers via gating, and (iii) linear-time computation with no covariance inversion.

Design principles. Method 1 is engineered around four constraints. The compute and memory consumption must remain essentially identical to the baseline so that any gain do not need extra computational resources because the original pipeline already needs over 24G graphic memory. Gates add only a shallow projection and a vector of weights, yielding $O(BD)$ overhead that matches the baseline encoder's per-batch cost. Additionally, gradients must stay stable on shallow, noisy features; gating plus a log-variance hinge dampens the influence of rare outliers and eliminates matrix inversions that become ill-conditioned. Moreover, the surrogate must respond directly to intra-class dispersion—the quantity that drives $\mathcal{H}(x|z)$—rather than to arbitrary clustering

artefacts. By tying the hinge threshold to the injected noise, the method explicitly asks whether variance has collapsed below the noise floor and only then pushes it up.

Expected behaviour and trade-offs. Because the hinge activates only when class variance undercuts the noise, the surrogate avoids over-regularizing naturally spread-out classes while still penalizing collapsed ones. The anticipated outcome is a modest accuracy drop (dispersion is increased) accompanied by higher reconstruction error (features are less recoverable). The surrogate is attacker-agnostic: it reshapes z so that any reasonable reconstructor must deal with elevated conditional entropy. It also preserves deployment footprint; no inference-time cost is added because gates are used only during training. This positions Method 1 as a "minimal intervention" that upholds the baseline threat model and architecture while improving the privacy–utility balance through a smoother, moment-based entropy proxy.

Method 1 is a controlled ablation of CEM: same cut, same attacker, same noise, but a surrogate that trades mixture expressiveness for robustness. Empirical results show that this restrained change yields the strongest privacy gains on default configurations .

Failure modes and tuning levers. Method 1 can underperform if the hinge threshold is set too low (privacy gradients vanish) or too high (over-regularization harms accuracy). Gate saturation can also freeze learning if initialization is too aggressive. To reduce the need for manual tuning, a conservative initialization (where gates start near-uniform) could be combined with a threshold that adapts to the injected noise level. Ablations planned later (varying τ, gate width, and loss scale) will probe these sensitivities within the same protocol.

3.3 Proposed Method 2: Slot + Gated Cross-Attention CEM

Method 2 explores richer structure without explicit density fitting. Building on slot attention [32], a small set of slots acts as learnable mixture components that iteratively aggregate class tokens via attention and GRU updates. Inspired by Flamingo's gated cross-attention [1], token–slot binding is modulated by gates that control how strongly slots rewrite token features. Slot-wise responsibilities, means, and variances drive a gated variance penalty that plays the role of L_C but with amortized mixture components.

Motivation and expectations. Slots offer a learnable analogue to mixture components: they can represent multiple modes within a class (e.g., pose, background, texture), and gated cross-attention can prevent early collapse by throttling how much slots alter token features. In theory, this combination should capture multi-modality that simple pooling might miss, potentially tightening the entropy surrogate when classes are inherently diverse.

Practical challenges. The added expressiveness comes with sensitivity. Slot count, attention temperature, and gate initialization all influence responsibility sharpness; poorly tuned values slow surrogate improvement or cause slot collapse. The optimization path is longer—slots must first stabilize before the variance penalty can effectively push dispersion. Under the default protocol, this manifested as stable accuracy but weaker privacy

than even the GMM baseline, indicating that variance remained under-regularized. Compute overhead also rises to $O(TBSD)$, which, while tractable on the target GPU, is still higher than Method 1.

Method 2 is treated as an exploratory branch whose insights feed a fusion design: pairing slots with the robust gated pooling (in parallel or via shortcut) to anchor variance while still modelling modes. It shows that expressiveness alone does not guarantee privacy gains under fixed resources and threat assumptions; stability and calibrated thresholds remain essential.

Failure analysis and levers. Early experiments suggest that responsibility sharpening, gate scales, and slot count jointly control stability. Too few slots collapse modes; too many diffuse responsibilities and mute the variance penalty. Similarly, if cross-attention gates start too large, tokens are rewritten aggressively and slots destabilize; if too small, slots never meaningfully bind. Planned ablations will vary slot count, temperature, and gate initializations while keeping the protocol fixed, guiding the eventual fusion with Method 1.

3.4 Evaluation Protocol

All variants are assessed under a single, locked protocol so that differences can only be attributed to the surrogate. Data and model: CIFAR-10 with standard normalization; VGG11-BN-SGM cut after layer 4; noise variance fixed at $\sigma = 0.025$. Optimization: SGD with momentum for 240 epochs, batch size 128, and a multi-step learning rate schedule (milestones 60/120/180/210/260, $\gamma = 0.2$) as in the reproduced CEM pipeline [52]. Threat model: identical white-box reconstructor trained on $\tilde{z}$ to convergence, with no auxiliary perceptual losses. Metrics: utility via top-1 accuracy; privacy via MSE, SSIM, and PSNR (standard image-quality metrics [22, 50]). Reproducibility: seeds, data order, and noise draws are fixed.

Fairness and scope. Architectural degrees of freedom (encoder depth, decoder capacity), cut position, noise schedule, and attacker strength are frozen; only the entropy surrogate changes. This prevents improvements from being confounded with deeper cuts or weaker attackers. Metrics are chosen to capture both goals of split learning: task fidelity (accuracy) and privacy (MSE/SSIM/PSNR), consistent with original pipeline. Runs share identical seeds and data ordering to reduce stochastic variance, and the attacker is always trained to convergence to avoid underestimating leakage. I kept the scope narrow on purpose (fixed dataset, cut and attacker) to make sure I could attribute any performance changes directly to the surrogate; future work can extend to additional datasets, alternative cuts, or black-box attackers once the surrogate behaviour is understood.

Ablation and reporting plan. Within this protocol, ablations will vary only surrogate-specific knobs (e.g., gate width and variance threshold for Method 1; slot count, attention temperature, and gate inits for Method 2) while keeping all else fixed. Results will be reported with paired accuracy and privacy metrics to highlight trade-offs, and, where resources permit, averaged over multiple seeds to expose variance. This keeps the methodological focus on the surrogate's causal effect rather than on incidental training noise.

Reproducibility and limits of current scope. Because the threat model is fixed and the codebase is deterministic (fixed seeds, fixed noise draws), experiments can be rerun to verify trends. However, the methodology consciously limits itself to one dataset and one cut position; it does not yet address distribution shift, larger images, or black-box attackers. These are deferred intentionally to keep the causal link between surrogate choice and experiment outcome clear; future extensions can expand the protocol to different configurations once the surrogate behaviour is fully characterized under this controlled setting.

4 Implementation

4.1 Architecture and Threat Model

The implementation keeps the collaborative split pipeline of Xia et al. [52] intact while replacing only the conditional-entropy surrogate. A VGG11-BN-SGM backbone [41] is cut after the fourth convolutional block, yielding an encoder $F_e : \mathbb{R}^{3\times32\times32} \to \mathbb{R}^{8\times8\times8}$ (flattened to $D = 512$) and a decoder F_d on the cloud. For a batch $\{(x_b, y_b)\}_{b=1}^{B}$,

$$z_b = F_e(x_b), \qquad \varepsilon_b \sim \mathcal{N}(0, \sigma^2 I), \qquad \tilde{z}_b = z_b + \varepsilon_b, \qquad g_b = F_d(\tilde{z}_b),$$

with task loss $L_D = \frac{1}{B}\sum_{b=1}^{B} \ell_{\mathrm{CE}}(g_b, y_b)$ [41]. The threat model fixes a strong white-box adversary: the attacker knows F_e, observes $\tilde{z}$, and trains a reconstructor A_ϕ to minimize $\|A_\phi(\tilde{z}) - x\|^2$ as in split-MIA settings [17,18]; privacy is reported with MSE/SSIM/PSNR [22, 50], and utility with top-1 accuracy. CIFAR-10 is normalized, noise variance is $\sigma = 0.025$, and no extra augmentation is used to avoid confounding the privacy effect. Architecture, cut position, and noise are held constant across all variants so that differences stem solely from the surrogate L_C.

4.2 Baseline Conditional-Entropy Surrogate

The original CEM optimizes $L = L_D + \lambda L_C$ with $\lambda = 16$, where L_C approximates $\mathcal{H}(x|z)$ via a k-component Gaussian mixture fitted on $\{\tilde{z}_b\}$ each epoch [52]. With mixture weights π_i, covariances Σ_i, and additive noise $\Sigma_p = \sigma^2 I$,

$$L_C^{\mathrm{GMM}} = \sum_{i=1}^{k} \pi_i \left(-\log \pi_i + \tfrac{1}{2} \log \frac{|\Sigma_i + \Sigma_p|}{|\Sigma_p|} \right).$$

Maximizing L_C therefore encourages dispersion of noisy features, which in turn raises a lower bound on reconstruction error. The proposed surrogates replace only this estimation while keeping the rest of the pipeline unchanged.

4.3 Training Schedule and Optimization

All variants follow a unified protocol (matching the reproduced original pipeline default): 240 epochs of SGD with momentum 0.9, weight decay 5×10^{-4}, batch size 128, and an initial learning rate of 0.05 with a multi-step schedule (milestones 60/120/180/210/260, $\gamma = 0.2$) [52]. Encoder and decoder are trained jointly; the attacker A_ϕ is trained after the classifier. Noise is injected once at the cut for both the task and the surrogate. Seeds, data order, and noise draws are fixed for reproducibility, following split-attack practice [17].

I kept the architecture, optimizer, and cut position fixed across all experiments. This ensures that any difference in results comes directly from the surrogate method itself, rather than from inconsistent training settings.

The attacker follows the original pipeline design proposed in CEM [52] : it receives $\tilde{z}$ and is trained with ℓ_2 reconstruction loss under the same normalization as the classifier [17,52]. No auxiliary perceptual losses are used, keeping the attacker a strong but bounded white-box reconstructor [25]. Training splits are identical across baselines and surrogates; convergence is monitored on the validation set to avoid underestimating the attacker, a standard precaution in white-box split attacks [18]. This alignment ensures that any change in reconstruction error arises from the encoder regularization rather than from attacker capacity [35].

4.4 Design Criteria for the Surrogate

The replacement for L_C must satisfy four constraints: (i) it remains fully differentiable and stable on shallow, noisy features [18]; (ii) it avoids iterative clustering to respect edge-side resource limits [10]; (iii) it responds to intra-class dispersion, which drives $\mathcal{H}(x|z)$ [52]; and (iv) it allows privacy gradients to be injected before task gradients, as in CEM [52]. Two surrogates were explored under these constraints: a gated-attention penalty that computes class-wise weighted moments, and a slot-based penalty that treats slots as learnable mixture components refined by gated cross-attention [1, 32].

4.5 Gated-Attention Surrogate

Gated moment estimation. For class c, define

$$Z_c = \{z_b : y_b = c\} \subset \mathbb{R}^{B_c \times D}.$$

where z_b denotes the encoder feature *before* the optional Gaussian noise injection at the cut (consistent with Algorithm 1 and the code implementation). Features are first normalized with LayerNorm, then passed through the gated attention mechanism of [23]:

$$\begin{aligned} V &= \tanh(W_V \,\mathrm{LN}(z_b)), \\ U &= \sigma(W_U \,\mathrm{LN}(z_b)), \\ \alpha_b &= \mathrm{softmax}_b\big(w^\top (V \odot U)\big), \end{aligned} \tag{1}$$

yielding softmax-normalized weights α_b per token. The class-wise weighted mean and variance are given by

$$\begin{aligned} \mu_c &= \frac{\sum_b \alpha_b \,\mathrm{LN}(z_b)}{\sum_b \alpha_b}, \\ v_c &= \frac{\sum_b \alpha_b \,(\mathrm{LN}(z_b) - \mu_c)^2}{\sum_b \alpha_b}. \end{aligned} \tag{2}$$

The low-rank projections W_V and W_U dampen noise, akin to efficient attention projections [5, 48].

Entropy-shaped penalty. The privacy term applies a hinge on the log-variance, with the threshold tied to the noise/regularization scale ($\texttt{var\_threshold} \times \texttt{reg\_strength}^2$):

$$\begin{aligned} L_c &= \frac{1}{D}\sum_{j=1}^{D} \max\Big(0,\, \log(v_{c,j}+\gamma) - \log\tau\Big), \\ \text{rob\_loss} &= \frac{\sum_{c\in\mathcal{C}_B} p_c\, L_c}{\sum_{c\in\mathcal{C}_B} p_c}, \qquad p_c = \frac{B_c}{B}, \\ L_C^{\text{gated}} &= s_{\text{cem}} \cdot \text{rob\_loss}, \qquad s_{\text{cem}} = \texttt{attention\_loss\_scale}, \\ L &= L_D + \lambda L_C^{\text{gated}}. \end{aligned} \tag{3}$$

In the implementation, $\gamma = 10^{-6}$ and $\tau = \max(\texttt{var\_threshold} \cdot \texttt{reg\_strength}^2, 10^{-8}) + \gamma$. Classes with $B_c \leq 1$ are skipped, hence the normalization by $\sum_{c\in\mathcal{C}_B} p_c$. Gradients propagate through W_V, W_U, and w into F_e, supplying dispersion-sensitive signals without clustering, similar in spirit to variance-floor regularizers used in noise-based defences [25]. The overall complexity is $O(\sum_c B_c D)$, and no matrix inversions are required [8].

Gradient injection order. Following the formulation in the original work, the overall objective can be written as

$$L = L_D + \lambda L_C^{\text{gated}},$$

where L_D denotes the task loss and L_C^{gated} represents the gated conditional-entropy surrogate. To ensure numerical stability during training, the implementation adopts a two-stage gradient computation scheme that is equivalent to this objective but differs in execution order.

Specifically, the privacy term L_C^{gated} (denoted as ℓ_{cem} in Algorithm 1) is first back-propagated with `retain_graph=True`, and the resulting gradients with respect to the encoder and gated-attention parameters are cached. Afterwards, gradients are cleared and the task-related objective used in implementation is back-propagated:

$$L_{\text{task}} = L_D + L_{\text{aux}},$$

where L_{aux} denotes optional auxiliary regularization terms, such as distance-correlation or GAN-based objectives when enabled. Before the optimizer update, the cached privacy gradients are added back to the encoder parameters, optionally scaled by the current learning rate, as described in Algorithm 1.

This staged gradient injection preserves the conceptual objective while improving optimization stability in early training, consistent with prior observations in collaborative inference and noise-based privacy regularization methods [17, 25].

Why gated attention helps. Gating smooths assignment noise, prevents a few outliers from dominating variance estimates, and keeps the surrogate differentiable everywhere. Unlike EM-fitting of GMMs, it introduces no discrete assignments and no covariance inversions. Because the hinge truncates gradients for compact classes, it avoids over-regularizing naturally tight clusters, which preserves accuracy.

Pseudo-code and data flow (Gated-Attention). Algorithm 1 restates the implemented Gated Attention CEM surrogate and its privacy-first update in a pseudocode form, and Figure 1 clearly illustrates the working principle of this proposed architecture.

Algorithm 1: Privacy-first update with the gated-attention CEM surrogate

Input: mini-batch (x, y); encoder F_e; head F_d; optimizer $\mathcal{O}$; scheduler LR η; epoch e; warmup $E_w = 5$; loss weight λ; regularization strength σ; `var_threshold` ρ; CEM scale $s_{\text{cem}} = 0.1$; flag `random_ini_centers`

Output: updated parameters of F_e, F_d, and gated-attention surrogate

$z \leftarrow F_e(x)$; $Z \leftarrow \text{vec}(z)$; $U \leftarrow \text{unique}(y)$; $B \leftarrow |x|$

$D \leftarrow \dim(Z)$

`use_att` $\leftarrow (\neg$`random_ini_centers`$) \wedge (\lambda > 0) \wedge (e > E_w)$

$\ell_{\text{cem}} \leftarrow 0$

if *use_att* ***and*** *Z has no NaN/Inf* **then**

  if *gated_attention_cem is uninitialized* **then**

    $h \leftarrow \min(512, \max(64, \lfloor D/4 \rfloor))$

    Initialize `gated_attention_cem` with (D, h, ρ, σ) and register its parameters in $\mathcal{O}$

  $\log \tau \leftarrow \log(\max(\rho \cdot \sigma^2, 10^{-8}) + \gamma)$ with $\gamma = 10^{-6}$

  `total` $\leftarrow 0$; `wgt` $\leftarrow 0$

  foreach $c \in U$ **do**

    $Z_c \leftarrow \{Z_b : y_b = c\}$; $M \leftarrow |Z_c|$

    if $M \leq 1$ **then**

      continue

    $X \leftarrow \text{LayerNorm}(Z_c)$

    $V \leftarrow \tanh(W_V X)$; $G \leftarrow \sigma(W_U X)$

    $\alpha \leftarrow \text{softmax}\big(w^\top (V \odot G)\big)$ over tokens

    $\mu \leftarrow \sum_m \alpha_m X_m$; $v \leftarrow \sum_m \alpha_m (X_m - \mu)^2$

    $v \leftarrow \max(v, \epsilon)$ with $\epsilon = 10^{-6}$

    $L_c \leftarrow \text{mean}_j \max\big(0, \log(v_j + \gamma) - \log \tau\big)$

    `total` $\leftarrow$ `total` $+ (M/B)\, L_c$; `wgt` $\leftarrow$ `wgt` $+ (M/B)$

  $\ell_{\text{cem}} \leftarrow s_{\text{cem}} \cdot$ `total`$/ \max($`wgt`$, 10^{-8})$

$\tilde{z} \leftarrow z$; **if** *Gaussian noise enabled* **then**

  $\tilde{z} \leftarrow z + \sigma \cdot \mathcal{N}(0, I)$

$g \leftarrow F_d(\tilde{z})$; $\ell_{\text{task}} \leftarrow \text{CE}(g, y)$

if *use_att* ***and*** *ℓ_{cem} requires grad* **then**

  Backprop ℓ_{cem}; cache encoder and surrogate gradients; $\mathcal{O}$.zero_grad()

Backprop ℓ_{task}

if *cached encoder gradients exist* **then**

  $s_\eta \leftarrow 1$ if $\eta < 4.1 \times 10^{-4}$ else $0.001/\eta$

  Merge cached privacy gradients into the encoder (scaled by $\lambda\, s_\eta$)

  Restore cached surrogate gradients

$\mathcal{O}$.step()

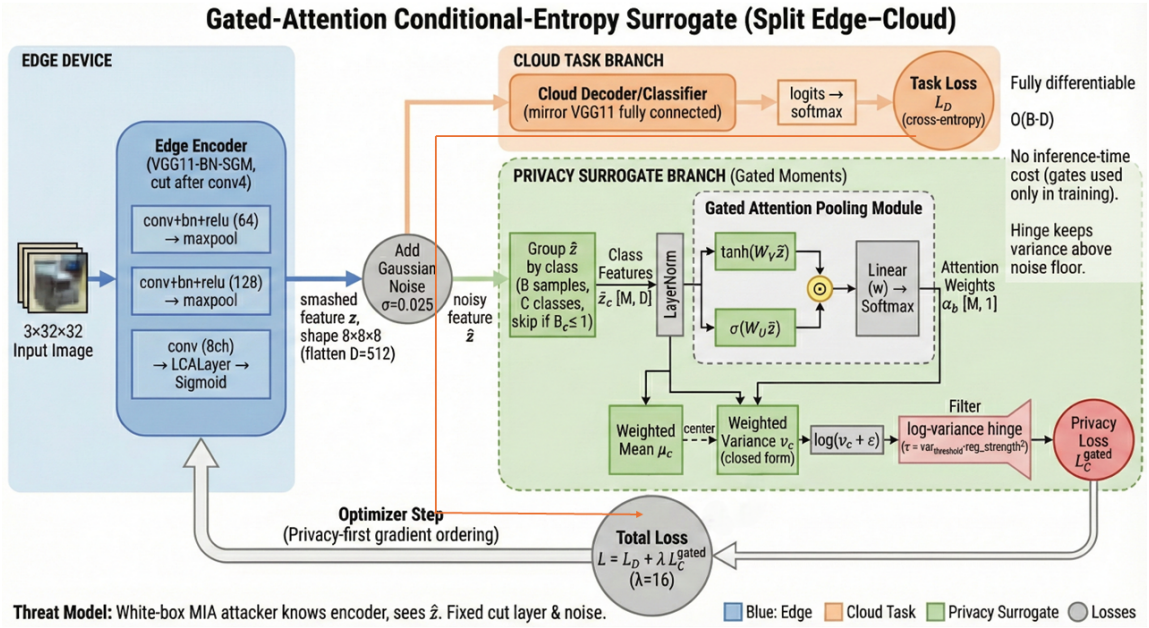

Figure 1: Gated Attention Architecture Diagram. This figure provides a conceptual overview; in the implementation, the privacy gradients are injected via a privacy-first two-stage backward pass (Algorithm 1).

4.6 Slot + Gated Cross-Attention Surrogate

Slot inference as learned mixtures. Slots $s^{(0)} \in \mathbb{R}^{S\times D}$ are initialized from a learned Gaussian $\mathcal{N}(\mu_\theta, \sigma_\theta^2 I)$ with trainable mean and log-variance. For class c, tokens $T_c \in \mathbb{R}^{B_c\times D}$ are taken from encoder features z *before* the optional Gaussian noise injection, then normalized and projected following slot-attention updates [32]:

$$\begin{aligned} \alpha^{(t)} &= \text{softmax}\left(\frac{T_c W_k (s^{(t-1)} W_q)^\top}{\sqrt{D}}\right), \\ u^{(t)} &= \alpha^{(t)\top}(T_c W_v), \\ s^{(t)} &= \text{GRU}\big(s^{(t-1)}, u^{(t)}\big) + \text{MLP}\big(s^{(t-1)}\big), \qquad t = 1, \ldots, T. \end{aligned} \tag{4}$$

LayerNorm and residuals are applied as in Slot Attention [32]. Temperature $D^{1/4}$ smooths assignments. Slots act as amortized mixture components that adapt to class-specific dispersion without explicit density estimation.

Gated cross-attention refinement. After slots are obtained from the iterative slot-attention updates, tokens query these fixed slots via multi-head cross-attention (four heads), and gates are applied on the cross-attention residuals (not on the slot updates themselves) to control how strongly slots rewrite token features [1, 45]:

$$\begin{aligned} \hat{T}_c &= T_c + \tanh(\alpha_{\text{xattn}})\,\text{CrossAttn}(T_c, S), \\ \tilde{T}_c &= \hat{T}_c + \tanh(\alpha_{\text{ffn}})\,\text{FFN}(\hat{T}_c). \end{aligned} \tag{5}$$

Here α_{xattn} and α_{ffn} are learned scalars initialized to 0.1 following Flamingo's stabilized gating [1]. Gating prevents assignment collapse and modulates the strength of slot-token binding in early epochs.

Dispersion and class penalty. Responsibilities r_{mk} are computed using cosine similarity with temperature β, following scaled dot-product attention [32, 45]:

$$r_{mk} = \mathrm{softmax}_k\big(\beta \cos(\mathrm{norm}(T_{c,m}), \mathrm{norm}(s_k))\big). \tag{6}$$

Slot-wise moments are defined as

$$\begin{aligned} \mu_k &= \frac{\sum_m r_{mk}\, \tilde{T}_{c,m}}{\sum_m r_{mk}}, \\ v_k &= \frac{\sum_m r_{mk}\, (\tilde{T}_{c,m} - \mu_k)^2}{\sum_m r_{mk}}. \end{aligned} \tag{7}$$

Two gates modulate variance contributions. The per-dimension gate is

$$g_{k,d} = \sigma\big(\mathrm{MLP}\big(\mathrm{LN}(\log v_k)\big)\big)\,,$$

and the signal-to-noise ratio (SNR) gate is

$$g_{\mathrm{snr},k,d} = \sigma(\beta_{\mathrm{snr}}(\mathrm{SNR}_{k,d} - t_{\mathrm{snr}}))\,. \qquad \mathrm{SNR}_{k,d} = \frac{v_{k,d}}{\mu_{k,d}^2 + \epsilon}.$$

A soft threshold is applied via

$$h_{k,d} = \frac{\mathrm{softplus}\big(\beta_s(\log v_{k,d} - \log \tau - m_0)\big)}{\beta_s + \epsilon}.$$

Slot masses are computed as

$$m_k = \sum_m r_{mk},$$

and are sharpened using an exponent p (default 2.5) to emphasize confident slots.

The resulting class penalty is

$$\begin{aligned} \ell_c &= g_c \cdot \frac{1}{D} \sum_{d=1}^{D} \sum_{k=1}^{S} \tilde{w}_{c,k}\, g_{k,d}\, g_{\mathrm{snr},k,d}\, h_{k,d}, \\ \tilde{w}_{c,k} &= \frac{(m_k/M)^p}{\sum_{k'} (m_{k'}/M)^p}, \qquad m_k = \sum_m r_{mk}, \\ g_c &= \sigma\big(a_c(p_c - b_c)\big), \qquad p_c = \frac{B_c}{B}, \\ \mathrm{rob\_loss} &= \frac{\sum_{c \in \mathcal{C}_B} p_c\, \ell_c}{\sum_{c \in \mathcal{C}_B} p_c}, \\ L_C^{\mathrm{slot}} &= s_{\mathrm{cem}} \cdot \mathrm{rob\_loss}, \qquad s_{\mathrm{cem}} = \texttt{attention\_loss\_scale}. \end{aligned} \tag{8}$$

Here, β_s and m_0 are learnable slope and margin parameters in the softplus hinge (a smooth approximation of ReLU) used to penalize log-variances below τ, while g_c and the class-weighted aggregation match the implementation's class-balanced surrogate.

The conceptual objective is

$$L = L_D + \lambda L_C^{\mathrm{slot}}.$$

Gradient injection order. Following the formulation in the original work, the overall objective can be written as

$$L = L_D + \lambda L_C^{\text{slot}}.$$

In the implementation, however, this objective is realized through a privacy-first two-stage backward procedure for improved stability during optimization.

Concretely, the privacy surrogate L_C^{slot} (denoted as ℓ_{cem} in Algorithm 2) is back-propagated first with `retain_graph=True`, and the resulting gradients with respect to the encoder and surrogate parameters are cached. The accumulated gradients are then cleared, and the task-related objective used in code is back-propagated:

$$L_{\text{task}} = L_D + L_{\text{aux}},$$

where L_{aux} collects optional auxiliary regularizers. Before the optimizer update, the cached privacy gradients are merged back into the encoder gradients, scaled by λ and an additional learning-rate-dependent factor, as specified in Algorithm 2.

Default configuration and sensitivity. The exploratory configuration uses $S = 8$, $T = 3$, four heads, $\alpha_{\text{xattn}} = \alpha_{\text{ffn}} = 0.1$, $p = 2.5$, and τ tied to σ^2 $[1, 32]$. Increasing S or T raises expressiveness but also instability and compute, matching observations in object-centric slot models [32]. Empirically, L_C^{slot} decreased slowly and privacy weakened relative to GMM, indicating that dispersion estimates were under-regularized; fusing slots with the simpler gated pooling (parallel or shortcut coupling) is the next step to stabilize variance [23].

Intuition for key knobs. The slot surrogate relies on token-to-slot assignments, so its behaviour is largely determined by how *sharp* these assignments are. If assignments are too soft, every slot receives almost the same mass (m_k becomes nearly uniform). If this happens, the statistics for each slot end up looking very similar. As a result, the loss term L_C^{slot} just acts like a weak average penalty. It fails to capture the true diversity within the class, which leads to weak privacy gradients. To avoid this, the exponent $p > 1$ is used to sharpen slot masses. However, p cannot be too large: it pushes almost all mass onto a single slot, which effectively recreates hard clustering and makes training sensitive and unstable.

The cosine temperature β offers another control knob: larger β makes the softmax assignments more peaked, while smaller β keeps them smoother. A moderate β helps avoid both extremes (uniform assignments vs. near one-hot collapse). Finally, the gated cross-attention scales α_{xattn} and α_{ffn} decide how strongly the slot pathway is allowed to modify token features. Initializing them small keeps the slot branch as a gentle residual refinement early on, so the encoder features are not aggressively rewritten before the slot assignments stabilize.

Pseudo-code and data flow (Slot + Gated Cross-Attention). Algorithm 2 restates the implemented Slot + Gated Cross Attention CEM surrogate and its privacy-first update in a pseudocode form, and Figure 2 clearly illustrates the working principle of this proposed architecture.

Algorithm 2: Privacy-first update with Slot + Gated Cross-Attention CEM surrogate

Input: mini-batch (x, y); encoder F_e; head F_d; optimizer $\mathcal{O}$; scheduler LR η; epoch e; warmup $E_w = 3$; loss weight λ; regularization strength σ; `var_threshold` ρ; CEM scale $s_{\text{cem}} = 0.25$; slots S=8, heads H=4, iterations T=3; flag `random_ini_centers`

Output: updated parameters of F_e, F_d, and SlotCrossAttentionCEM surrogate

$z \leftarrow F_e(x)$; $Z \leftarrow \text{vec}(z)$; $U \leftarrow \text{unique}(y)$; $B \leftarrow |x|$

$D \leftarrow \dim(Z)$

`use_att` $\leftarrow (\neg$`random_ini_centers`$) \wedge (\lambda > 0) \wedge (e > E_w)$

$\ell_{\text{cem}} \leftarrow 0$

if *`use_att` **and** Z has no NaN/Inf* **then**

  if *`attention_cem` is uninitialized* **then**

    Initialize `attention_cem` with $(D, S{=}8, H{=}4, T{=}3, \epsilon_{\text{var}}{=}10^{-4}, \rho, \sigma)$ and register its parameters in $\mathcal{O}$

  $\tau \leftarrow \max(\rho \cdot \sigma^2, 10^{-8}) + \gamma$ with $\gamma = 10^{-6}$

  `total` $\leftarrow 0$; `wgt` $\leftarrow 0$

  foreach $c \in U$ **do**

    $Z_c \leftarrow \{Z_b : y_b = c\}$; $M \leftarrow |Z_c|$

    if $M \leq 2$ **then**

      continue

    $X \leftarrow \text{LayerNorm}(Z_c)$

    $s^{(0)} \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2 I)$

    $S_c \leftarrow \text{SlotAttn}(X, s^{(0)}; T)$

    $\tilde{X} \leftarrow \text{CrossAttn}(X, S_c; \tanh(\alpha_{\text{xattn}}), \tanh(\alpha_{\text{ffn}}))$

    $r \leftarrow \text{softmax}\big(\beta_a \cos(\text{norm}(X), \text{norm}(S_c))\big)$

    $m \leftarrow \sum_m r_{m,:} + \epsilon$; $\mu_s \leftarrow (r^\top \tilde{X})/m$; $v_s \leftarrow \sum_m r_{m,:}(\tilde{X}_m - \mu_s)^2/m$

    $v_s \leftarrow \max(v_s, \epsilon_{\text{var}})$; $\log v_s \leftarrow \log(v_s + \gamma)$

    $g_d \leftarrow \sigma(\text{MLP}(\text{LN}(\log v_s)))$

    $\text{snr} \leftarrow v_s/(\mu_s^2 + \epsilon)$; $g_{\text{snr}} \leftarrow \sigma(\beta_{\text{snr}}(\text{snr} - t_{\text{snr}}))$

    $h \leftarrow \text{softplus}\big(\beta_s(\log v_s - \log \tau - m_0)\big)/(\beta_s + \epsilon)$

    $q \leftarrow g_d \odot g_{\text{snr}} \odot h$

    $w \leftarrow \text{norm}\big((m/M)^{p_s}\big)$; $\ell_c \leftarrow \text{mean}_d \sum_s w_s q_{s,d}$

    $g_c \leftarrow \sigma(a_c(M/B - b_c))$

    $\ell_c \leftarrow g_c \, \ell_c$

    `total` $\leftarrow$ `total` $+ (M/B)\, \ell_c$; `wgt` $\leftarrow$ `wgt` $+ (M/B)$

  $\ell_{\text{cem}} \leftarrow s_{\text{cem}} \cdot$ `total`$/\max($`wgt`$, 10^{-8})$

$\tilde{z} \leftarrow z$; **if** *`Gaussian noise enabled`* **then**

  $\tilde{z} \leftarrow z + \sigma \cdot \mathcal{N}(0, I)$

$g \leftarrow F_d(\tilde{z})$; $\ell_{\text{task}} \leftarrow \text{CE}(g, y)$

if *`use_att` **and** ℓ_{cem} requires grad* **then**

  Backprop ℓ_{cem}; cache encoder and surrogate gradients; $\mathcal{O}$.zero_grad()

Backprop ℓ_{task}

if *cached encoder gradients exist* **then**

  $s_\eta \leftarrow 1$ if $\eta < 4.1 \times 10^{-4}$ else $0.001/\eta$

  Merge cached privacy gradients into the encoder (scaled by $\lambda\, s_\eta$)

  Restore cached surrogate gradients

$\mathcal{O}$.step()

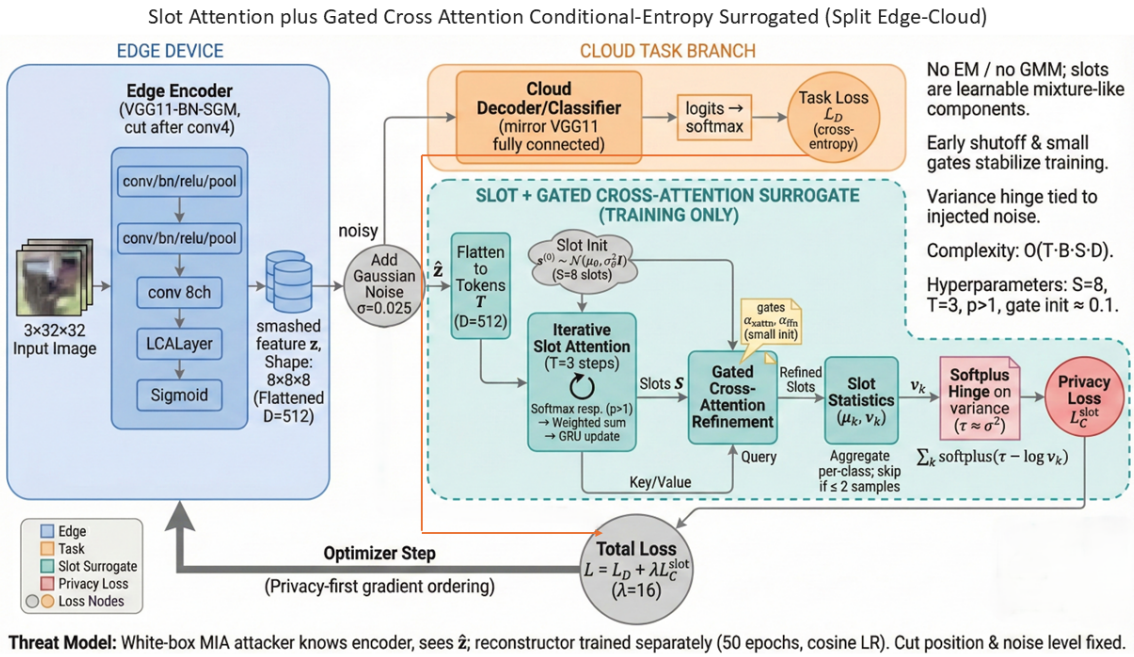

Figure 2: Slot Attention + Gated Cross Attention Architecture Diagram. This figure provides a conceptual overview; in the implementation, the privacy gradients are injected via a privacy-first two-stage backward pass (Algorithm 2).

4.7 Numerical Stability and Regularization

Both surrogates share safeguards: $\epsilon = 10^{-6}$ in variances and logs; skipping classes with too few samples (e.g., $B_c \leq 1$ for gated, $B_c \leq 2$ for slots); NaN/Inf checks that zero the surrogate loss while keeping the graph intact; and a short warmup (three epochs in current code) before activating the surrogate [18]. In addition to epoch warmup, the slot surrogate applies an early shutoff that returns a zero L_C for the first N forward calls (and when gate statistics indicate overly broad activation), to suppress unstable privacy gradients while keeping the computation graph connected for consistent training logic. Hinge and softplus thresholds preserve gradients near decision boundaries, avoiding dead zones [25]. Class-balanced normalization prevents frequent labels from dominating L_C, echoing class-balancing practices in split-MIA evaluations [17]. Exponential moving averages or gradient clipping are potential future stability tweaks if needed.

4.8 Algorithmic Integration

Each training step follows a deterministic privacy-first order. For the gated surrogate:

1. Forward: $z = F_e(x)$, $\tilde{z} = z + \varepsilon$, logits $g = F_d(\tilde{z})$.
2. Compute L_D; group z (pre-noise) by class; compute L_C^{gated} from gated moments.
3. Backpropagate L_C^{gated}; save encoder and gate gradients.
4. Zero gradients; backpropagate L_D.

5. Add saved gradients (optionally scaled by the current learning rate) to encoder parameters; take the optimizer step.

The slot surrogate uses the same scaffold with L_C^{slot}. This schedule matches CEM's gradient ordering, eliminates per-epoch clustering, and keeps inference unchanged [17,52].

4.9 Computation and Memory Profile

The gated surrogate adds $O(Dh)$ parameters (projection plus gate vector) and $O(BD)$ flops; memory overhead is negligible [20]. The slot + gated cross-attention surrogate adds $O(SD)$ parameters, $O(TBSD)$ flops, and $O(BS)$ attention memory; with $S = 8, T = 3$ it remains tractable on RTX 5880 Ada but is heavier than gated pooling [32]. Neither surrogate affects deployment cost: L_C is used only in training, and the encoder–decoder graph at inference matches the baseline footprint [18].

4.10 Hyper-parameters and Planned Ablations

Gated-attention axes: projection width h, threshold τ relative to σ^2, gate initialization scale, and loss weight λ (fixed at 16 for fair comparison) [23]. Slot axes: number of slots S, iterations T, head count, gate initializations $\alpha_{\text{xattn}}, \alpha_{\text{ffn}}$, sharpening power p, temperature β, and class gates [1, 32]. Planned ablations include (i) sweeping σ and τ jointly to calibrate the hinge/softplus boundary; (ii) removing SNR gates to test their contribution; (iii) varying p to control slot mass sharpening; and (iv) fusing slots with gated pooling in parallel or via shortcut to stabilize variance while retaining multimodal capacity [45].

4.11 Reproducibility Controls

To keep results attributable to the surrogate, the following are fixed across runs: backbone, cut layer, noise level, optimizer, schedule, batch size, and attacker architecture [17, 18]. Seeds, data order, and noise sampling are deterministic [25]. Under these controls, the observed pattern is consistent: gated attention improves privacy with a modest accuracy drop, whereas slot + gated cross-attention remains competitive on accuracy but weakens privacy under the default configuration.

5 Experiments and Results Analysis

5.1 Experimental Setup

All three runs follow the fixed CIFAR-10 split-learning protocol: VGG11-BN-SGM cut after layer 4, smashed-data bottleneck 8 channels, additive Gaussian noise $\sigma = 0.025$, loss weight $\lambda = 16$, batch size 128, SGD with a multi-step LR schedule (milestones 60/120/180/210/260[1], $\gamma = 0.2$) for 240 epochs on a single NVIDIA RTX 5880 Ada. The white-box MIA reconstructor (Adam, 50 attack epochs with cosine-annealed LR) consumes the same $\tilde{z}$ and is evaluated on the target client, consistent with split-attack practice. Cut position, noise, optimizer, and attacker are identical across methods to isolate the surrogate effect, following best-practice fairness guidelines for comparative robustness studies [17, 18, 25].

[1]Milestone 260 is kept only for runs longer than 240 epochs; it is inactive here.

Scripts and key knobs. Baseline: $\lambda = 16$, regularization strength 0.025, SCA_new adversarial regularizer 0.3, bottleneck option `noRELU_C8S1`, var_threshold 0.125, MultiStep LR from 0.05, seed 125. Gated-attention run inherits the same backbone/cut/noise and keeps $\lambda = 16$ (log shows warmup at $\lambda \approx 8$ then $\lambda = 16$ after LR decay). Slot + gated cross-attention keeps $\lambda = 16$ as well to remain comparable to the default script. All runs use the same attack configuration: 50 attack epochs, GAN auto-encoder `res_normN8C64` [15], attack loss default MSE unless specified otherwise, average_time=1.

Role of auxiliary knobs. `SCA_new` adversarial regularizer (0.3) is the same defence term used in the published baseline; it adversarially shapes smashed features against an auxiliary discriminator [25] and is held fixed to isolate the effect of the entropy surrogate. The bottleneck option `noRELU_C8S1` reduces channel count to 8 without an extra ReLU, matching the baseline's split footprint and bandwidth; changing it would alter the attack surface, so it is frozen for comparability [18]. The variance threshold 0.125 appears in both the baseline and surrogates as the target floor for smashed-data dispersion; for gated attention it aligns the hinge to the noise level, and for slots it sets the softplus threshold.

Training/attack timeline. Classification training spans 240 epochs; attack training spans 50 epochs per method, matching prior attacker-strength baselines [17]. The logs show per-epoch Prec@1 progression and attack-side prediction accuracy progression. Inference-time footprint is identical because L_C is used only during training; the deployed encoder–decoder equals the baseline [52].

Runtime/throughput. On RTX 5880 Ada, one classifier epoch costs around 8–10 s (depending on phase and surrogate); attack epochs cost around 1.2–1.9 s for feature inference plus around 0.25–0.38 s for surrogate statistics. These timings confirm the surrogates introduce negligible overhead relative to the baseline training loop [25].

5.2 Overall Results

Utility and privacy metrics are summarized in Figure 3. Baseline (GMM) reaches 85.18% top-1 and MIA MSE/SSIM/PSNR of 0.0436 / 0.432 / 13.60. Gated attention achieves 84.34% accuracy with improved privacy (0.0473 / 0.411 / 13.25), reflecting the effect of variance-floor gating seen in obfuscation defences [25]. Slot + gated cross-attention attains 84.69% accuracy but weaker privacy (0.0393 / 0.459 / 14.06), consistent with instability reports for mixture-like surrogates on shallow features [32]. Under identical threat and training conditions, the gated surrogate offers the best privacy–utility trade-off; the slot branch retains slightly higher accuracy than gated attention but leaks more [17,18]. SSIM/PSNR follow the standard image quality definitions in [50], and the privacy trend (higher MSE, lower SSIM) matches expectations from stronger regularization in split defences [10, 25].

Method	**Prec@1 (%)**	**MSE ↑**	**SSIM ↓**	**PSNR (dB) ↓**
Baseline (GMM)	**85.18**	0.0436	0.432	13.60
Gated attention	84.34	**0.0473**	**0.411**	**13.25**
Slot + gated cross-attn	84.69	0.0393	0.459	14.06

Table 1: Summary of utility and privacy metrics on CIFAR-10 (fixed cut/noise/attacker). Gated attention yields the strongest privacy (MSE ↑, SSIM/PSNR ↓) with only a modest accuracy drop.

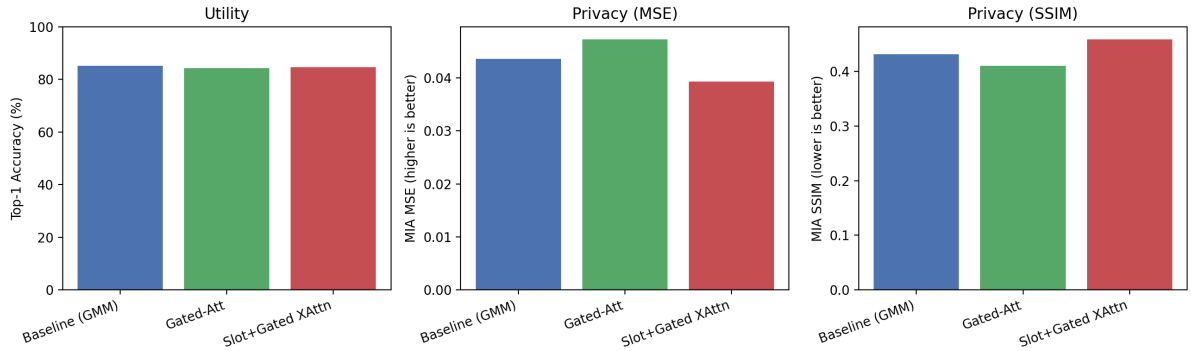

Figure 3: Utility and privacy comparison on CIFAR-10 (higher MSE, lower SSIM indicate stronger privacy). Hardware: RTX 5880 Ada; identical cut/noise/attacker across runs.

5.3 Attack Training Dynamics

Figure 4 plots the MIA decoder's prediction accuracy over 50 attack epochs. All methods converge to mid-20% accuracy, consistent with noisy smashed data [17, 18]. Gated attention does not slow attack convergence but still raises reconstruction error (higher MSE, lower SSIM), showing that dispersion—not just confusing the attack classifier—drives its privacy gain [25, 52]. The slot variant shows slightly lower attack accuracy early on yet leaks more (lower MSE, higher SSIM), indicating insufficient variance inflation in the slot branch, a known challenge for unstable mixture-like surrogates [10, 32].

5.4 Training Trajectory and Checkpoints

Across 240 classifier epochs, each method saves a best checkpoint (the script's `best` model) which is then used for MIA evaluation. Under this protocol, the achieved Prec@1 is 85.18% for the baseline, 84.34% for gated attention, and 84.69% for slot + gated cross-attention, consistent with the stability expected from the fixed multi-step schedule in the reproduced pipeline [41, 52]. Attack-side prediction accuracy curves plateau around 24–26% for gated/slot and ∼24% for baseline, consistent with prior split-attack training curves [17, 18]. The decisive signal comes from reconstruction metrics: gated attains the highest MSE (0.0473) and lowest SSIM (0.4109), baseline sits in the middle (0.0436 / 0.4316), and slot lags (0.0393 / 0.4593). PSNR follows the same ordering (gated 13.25 dB < baseline 13.60 dB < slot 14.06 dB), reinforcing that gated attention makes reconstructions visually worse despite similar attack accuracy [25, 50, 52].

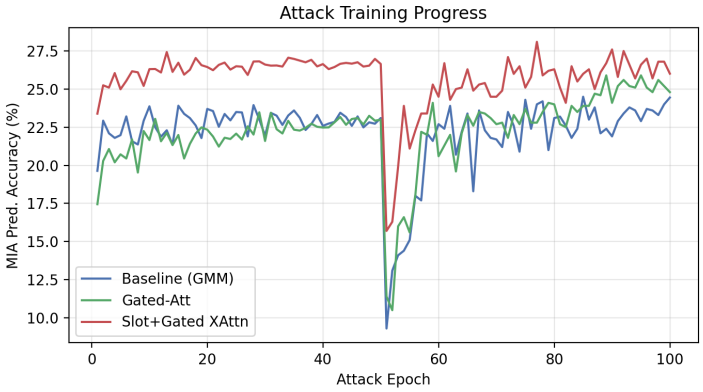

Figure 4: MIA prediction accuracy during attack training (lower is better).

5.5 Failure Analysis on Slot Branch

The logs reveal that even under the default entropy weight ($\lambda = 16$), slot responsibilities sharpened slowly: responsibilities and slot masses remain diffuse, so per-slot variances v_k are not pushed above the noise floor [32]. Cosine-temperature and gate initializations (cross-attn gates at 0.1) may be too conservative, leading to weak token–slot binding early on; conversely, increasing temperature without anchoring variance risks slot collapse [45]. The SNR and variance gates stay near the linear regime, so the softplus threshold rarely fires, limiting gradient magnitude on v_k [1]. Together these factors explain why SSIM rises (weaker privacy) despite reasonable attack accuracy: dispersion is not sufficiently inflated. The planned fusion with gated pooling (parallel/shortcut) is intended to inject a stronger variance floor while slots capture modes, mitigating these failure modes [23].

5.6 Additional Visualizations

To further illustrate privacy signals, Figure 5 shows PSNR (lower is better privacy), and Figure 6 plots the accuracy–privacy trade-off (MSE/SSIM vs top-1). As shown in the plot, the Gated Attention method achieves the best trade-off: it improves privacy (higher MSE, lower SSIM) while maintaining an accuracy very close to the baseline; the slot variant maintains accuracy levels similar to the baseline, but its privacy protection is weaker. This trade-off is quite common in other studies on obfuscation [10, 25].

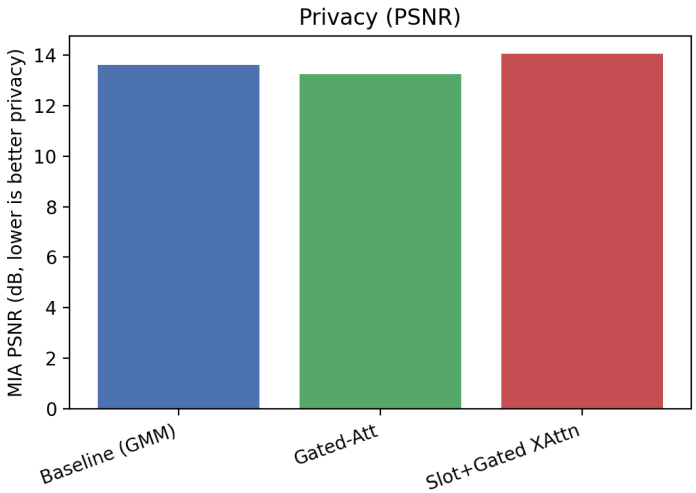

Figure 5: Privacy via PSNR (dB, lower is better). Gated attention yields the lowest PSNR, indicating harder reconstructions for the attacker.

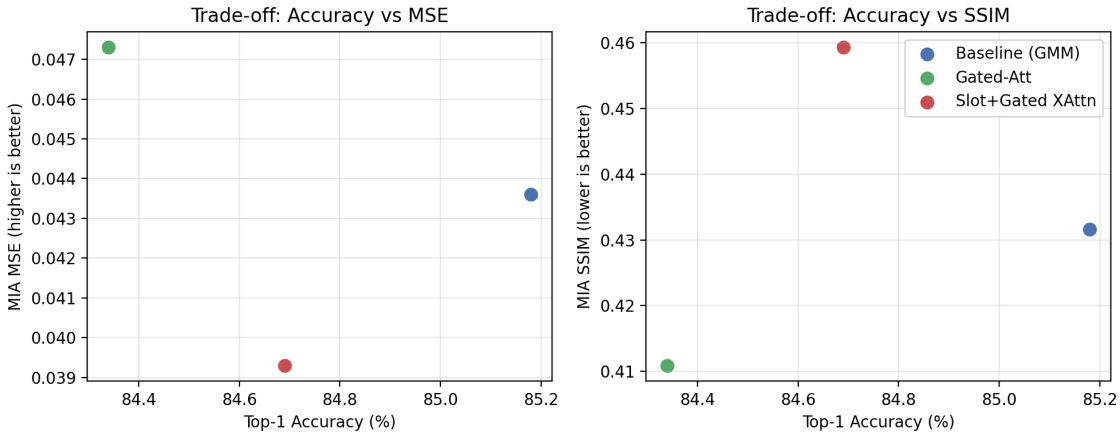

Figure 6: Accuracy–privacy trade-off. Left: accuracy vs MSE (higher is better privacy); Right: accuracy vs SSIM (lower is better privacy). Gated attention moves up/right in MSE space and down in SSIM with a modest accuracy loss; slot remains closer to baseline on accuracy than gated attention but loses privacy.

5.7 Takeaways and Next Steps

Across identical scripts and hardware, Method 1 (gated attention) is the only surrogate that strengthens privacy with a modest utility drop, consistent with the stabilizing role of gates in prior attention modules [20, 51]. Method 2 highlights that additional expressiveness alone does not guarantee robustness without stable variance control [32]. Immediate next steps: sweep slot/gate hyper-parameters, fuse slots with gated pooling (parallel/shortcut) to stabilize dispersion, and extend evaluation to additional cuts/datasets to test generality, following evaluation practices in split defences [18, 25].

6 Challenges Reflection and Progress Management

In the proposal, the near-term milestones before the interim deadline were: (i) establish a clear threat model and literature baseline for collaborative inference and model inversion; (ii) reproduce Conditional Entropy Maximization (CEM) under the default protocol; and (iii) implement and benchmark at least one alternative surrogate of $\mathcal{H}(x|z)$ under a fixed attacker [17, 18, 52]. The longer-term objective remains unchanged: deliver a distribution-agnostic surrogate that improves the privacy–utility trade-off over CEM across four datasets under comparable split and compute constraints [10, 25, 52].

6.1 Progress Against Workplan

Completed deliverables (with measurable outcomes). Baseline CEM reproduction on CIFAR-10 has been completed using the default experimental script in the original pipeline source code provided by authors[2] [52], matching the expected utility and privacy levels (Prec@1 85.18%, MSE 0.0436, SSIM 0.432, PSNR 13.60), and this gives me a reliable reference point for comparing my proposed methods later on. Building on this baseline, two attention-based surrogates of conditional entropy were implemented and evaluated under identical training and attacker budgets (240 classifier epochs; 50 white-box attack epochs with fixed scripts and schedules) on a single RTX 5880 Ada [17]. Method 1 (gated attention) improves privacy relative to the baseline (MSE 0.0473 ↑, SSIM 0.411 ↓, PSNR 13.25 dB ↓) with a small utility drop (84.34%), whereas Method 2 (slot + gated cross-attention) achieves competitive utility (84.69%) but underperforms on privacy (MSE 0.0393 ↓, SSIM 0.459 ↑), motivating the fusion direction described below [1, 20, 32].

Planned milestones vs current status. Table 2 and Figure 7 summarize planned vs achieved work packages up to the interim deadline. The main deviation is not a schedule slip but an outcome mismatch: the slot-based branch consumed the intended implementation window yet did not translate additional expressiveness into higher conditional entropy under the fixed threat model, so the next phase prioritizes fusion and ablations that preserve the gated baseline as a stable anchor [17, 25].

[2] https://github.com/xiasong0501/CEM

Work package	Planned	Measurable Outcome	Status
Literature review & threat model	Sep–Nov 2025	Surveyed split inference and MIAs; curated bibliography and threat model used throughout the report	Completed
Baseline reproduction	Oct–Nov 2025	CIFAR-10 reproduction: Prec@1 85.18%, MSE 0.0436, SSIM 0.432, PSNR 13.60	Completed
Method 1: gated-attention surrogate	Nov–Dec 2025	Strongest privacy among tested variants: MSE 0.0473, SSIM 0.411, PSNR 13.25; Prec@1 84.34%	Completed
Method 2: slot + gated cross-attention	Nov–Dec 2025	Utility competitive (84.69%) but privacy weaker (MSE 0.0393, SSIM 0.459); failure analysis drafted	Completed
Fusion & ablations (integration)	Dec 2025–Feb 2026	In progress: parallel/shortcut fusion prototypes and controlled ablations under the fixed attacker	In progress
Cross-dataset evaluation (4 datasets)	Jan–Apr 2026	Planned: extend protocol beyond CIFAR-10 (e.g., SVHN, FaceScrub) with identical cut/noise/attacker	Planned

Table 2: Progress against the proposal workplan up to the interim deadline, anchored by measurable outcomes.

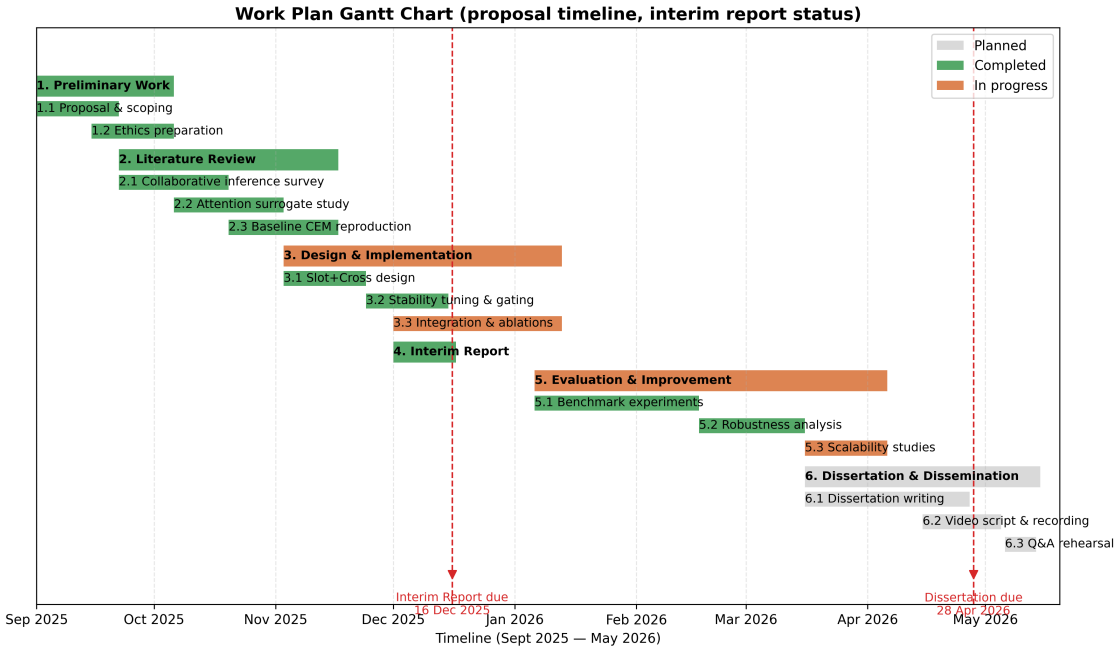

Figure 7: Updated proposal-style Gantt chart showing completed (green), in-progress (orange), and planned (gray) work packages up to the interim deadline.

6.2 Challenges, Risks, and Mitigation

A major technical challenge is that shallow smashed features are simultaneously high-dimensional and noise-perturbed, making mixture fitting and mixture-like responsibilities

brittle [32, 52]. The slot branch appears under-regularized in this setting: it can preserve discriminative structure for the main task, yet dispersion control is not consistently activated, leading to lower reconstruction error (MSE ↓) despite similar attack-side prediction accuracy [17]. This motivates the fusion plan where gated pooling provides a stable variance floor while slots contribute controlled multi-modality [1, 23].

Below, I identify the main risks to the project schedule and how I plan to mitigate them. First, fusion may remain unstable and delay cross-dataset evaluation; mitigation is to prioritize a parallel fusion that retains the gated baseline as a safe fallback, and to stage ablations that remain informative even if fusion underperforms (e.g., varying thresholds, gate width, and loss scales) [16, 21]. Second, current conclusions may be CIFAR-10-specific; mitigation is to stage additional datasets sequentially (starting with SVHN and FaceScrub) using the frozen protocol, reporting deltas relative to CIFAR-10 to detect regressions early [18, 38, 39]. Third, privacy gains may depend on attacker design; mitigation is to include at least one stronger reconstructor (e.g., perceptual-loss decoder) while keeping the white-box assumption unchanged [17, 26]. Progress will be monitored with the same measurable criteria used so far: Prec@1 for utility, and MSE/SSIM/PSNR for privacy, with all comparisons made under identical budgets and scripts [22, 50].

6.3 Next Steps

Fusion design and verification (Dec 2025–Jan 2026). The priority is to fuse slots with gated pooling so that slots model multi-modality while gated pooling enforces a variance floor. Two variants will be implemented: (i) parallel fusion (slots and gated pooling computed in parallel and aggregated by learned weights); (ii) shortcut/residual fusion where slot dispersion is injected as a residual into the gated branch, leveraging residual couplings that stabilize deep networks [16, 21]. Both variants will be rerun on CIFAR-10 under the fixed white-box attacker to test whether privacy improves without harming accuracy [20, 23, 32].

Ablations and stability sweeps (Jan–Feb 2026). Planned sweeps include slot count, assignment temperature, gate scales, softplus margin/slope, and fusion weights, plus removal of SNR gates to isolate their effect. These will be evaluated under the same protocol to map stability/expressiveness trade-offs and to identify robust defaults for later datasets [1, 16, 45].

Composing defences under a fixed threat model (Feb–Mar 2026). Test whether combining the fused surrogate with lightweight noise/adversarial representation learning [25] or distance-correlation penalties [47] yields further privacy gains, while keeping cut/noise/attacker fixed to avoid confounds. Pruning-style regularization [10] will be explored only if compute allows, with fusion designs chosen to keep the budget consistent with edge constraints [18].

Cross-dataset evaluation toward the four-dataset target (Mar–Apr 2026). Extend evaluation beyond CIFAR-10 (e.g., SVHN and FaceScrub first) with the same cut/noise/attacker to test robustness to domain shift. Deltas relative to CIFAR-10 will be reported for each surrogate to assess transferability and to prioritize knobs that generalize [18, 38, 39].

Attacker variants and robustness checks (Apr 2026). Add one stronger attacker (e.g., perceptual-loss decoder [26]) to ensure improvements are not tied to a single loss, while keeping the white-box assumption fixed [17]. If needed, a generative-prior variant can be tested (e.g., GAN-based) to probe sensitivity to stronger priors [15].

7 Conclusion

This report investigated replacing the Gaussian-mixture surrogate of $\mathcal{H}(x|z)$ in CEM with attention-based, distribution-agnostic alternatives under a fixed split-learning threat model. Reproducing the baseline and two surrogates shows that gated attention is the only variant that materially improves privacy (MSE ↑, SSIM/PSNR ↓) while keeping top-1 accuracy within about 1% of the reproduced baseline (85.18% vs. 84.34%), whereas the exploratory slot + gated cross-attention remains competitive on accuracy (84.69%) but under-regularizes dispersion. The findings support the hypothesis that stable, moment-based gating is more reliable than EM fitting or unconstrained slots on shallow, noisy features, and they highlight the need for variance anchoring when modelling multi-modality. Immediate priorities are to fuse slots with gated pooling (parallel/shortcut or residual coupling), sweep fusion-specific hyper-parameters, and extend evaluation to additional datasets/attackers to confirm generality under identical architectural and noise budgets. With a standardized protocol, deterministic scripts, and updated Gantt plan in place, the project is on track to deliver a fused surrogate and broader evaluation by the final submission.

Methodologically, the work kept architecture, cut position, noise variance, attacker strength, and training schedule fixed, so observed deltas can be causally attributed to the entropy surrogate rather than to other changes. Privacy was assessed with MSE/SSIM/PSNR following standard image-quality metrics, and fairness was enforced by identical attack scripts across runs. So far, I have achieved three main outcomes in this project. First, the CEM baseline was faithfully reproduced under the published cut/noise/optimizer settings, establishing a trustworthy reference for privacy–utility trade-offs on CIFAR-10. Second, a gated-attention surrogate replaced GMM fitting with differentiable, class-wise variance control, yielding the best privacy metrics to date (MSE 0.0473, SSIM 0.411, PSNR 13.25) with only a minor Prec@1 drop relative to baseline (84.34% vs. 85.18%), and doing so without increasing inference cost. Third, an exploratory slot + gated cross-attention surrogate remained competitive on accuracy (84.69%) but exposed the limits of unconstrained mixture-like modelling on shallow features (MSE 0.0393, SSIM 0.459), motivating the forthcoming fusion design.

There are three main risks I need to manage. First, the slot branch lacks a variance floor, so I need to anchor the variance to stop privacy leakage. Second, since I only used CIFAR-10, I need to test on other datasets to ensure the results aren't overfitting. Finally, I need to re-check the variance thresholds for each surrogate to make sure the privacy regularization is working correctly. The updated Gantt reflects completed literature/baseline/gated milestones, in-progress fusion/ablations, and planned cross-dataset evaluations, aligning deliverables with the original proposal timeline.

Looking ahead, the fusion of slots with gated pooling will test whether classification utility and privacy can be combined within similar training budget. Extending the fixed-threat protocol to additional datasets and attacker variants will probe generality, while ablations on τ, gate scales, and slot temperature will map the stability landscape. These steps aim to produce a deployable, attacker-agnostic surrogate that surpasses the original baseline on both privacy and utility under realistic edge–cloud constraints.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikołaj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS 2022)*, 2022.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yere Yere. Neural machine translation by jointly learning to align and translate. *ArXiv*, 1409, 09 2014.

[3] Iz Beltagy, Matthew Peters, and Arman Cohan. Longformer: The long-document transformer, 04 2020.

[4] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5253–5270. USENIX Association, 2023.

[5] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations (ICLR)*, 2021.

[6] Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 2286–2296, 2021.

[7] Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 933–941, 2017.

[8] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–38, 1977.

[9] Benedikt Dietich, Nadja Peters, Sangyoung Park, and Samarjit Chakraborty. Estimating the limits of cpu power management for mobile games. In *2017 IEEE International Conference on Computer Design (ICCD)*, pages 1–8, 2017.

[10] Shiwei Ding, Lan Zhang, Miao Pan, and Xiaoyong Yuan. Patrol: Privacy-oriented pruning for collaborative inference against model inversion attacks. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4704–4713, 2024.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold,

Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[12] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4829–4837, 2016.

[13] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15)*, pages 1322–1333. Association for Computing Machinery, 2015.

[14] Xueluan Gong, Ziyao Wang, Shuaike Li, Yanjiao Chen, and Qian Wang. A GAN-based defense framework against model inversion attacks. *IEEE Transactions on Information Forensics and Security*, 18:4475–4487, 2023.

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS 2014)*, 2014.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[17] Zecheng He, Tianwei Zhang, and Ruby B. Lee. Model inversion attacks against collaborative inference. In *Proceedings of the 35th Annual Computer Security Applications Conference (ACSAC '19)*, pages 148–162, 2019.

[18] Zecheng He, Tianwei Zhang, and Ruby B. Lee. Attacking and protecting data privacy in edge–cloud collaborative inference systems. *IEEE Internet of Things Journal*, 8(12):9706–9716, 2021.

[19] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the gan: Information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)*, pages 603–618. Association for Computing Machinery, 2017.

[20] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7132–7141, 2018.

[21] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.

[22] Q. Huynh-Thu and M. Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics Letters*, 44:800–801, 2008.

[23] Maximilian Ilse, Jakub M. Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[24] 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 (ICML)*, pages 4651–4664, 2021.

[25] Jonghu Jeong, Minyong Cho, Philipp Benz, and Tae-hoon Kim. Noisy adversarial representation learning for effective and efficient image obfuscation. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI 2023)*, 2023.

[26] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 694–711, 2016.

[27] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 3744–3753, 2019.

[28] Jingtao Li, Adnan Siraj Rakin, Xing Chen, Zhezhi He, Deliang Fan, and Chaitali Chakrabarti. Ressfl: A resistance transfer framework for defending model inversion attack in split federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1913–1922, 2022.

[29] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

[30] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven C. H. Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *Advances in Neural Information Processing Systems (NeurIPS 2021)*, 2021.

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021.

[32] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems (NeurIPS 2020)*, 2020.

[33] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems (NeurIPS 2019)*, 2019.

[34] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference*

on Empirical Methods in Natural Language Processing (EMNLP), pages 1412–1421, 2015.

[35] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5188–5196, 2015.

[36] Charles H. Martin and Michael W. Mahoney. Traditional and heavy-tailed self regularization in neural network models. *ArXiv*, abs/1901.08276, 2019.

[37] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 739–753, 2019.

[38] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[39] Hong-Wei Ng and Stefan Winkler. A data-driven approach to cleaning large face datasets. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 343–347, 2014.

[40] Nir Shlezinger, Erez Farhan, Hai Morgenstern, and Yonina C. Eldar. Collaborative inference via ensembles on the edge. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8478–8482, 2021.

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 09 2014.

[42] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[43] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, 2019.

[44] Chandra Thapa, M. A. P. Chamikara, Seyit Camtepe, and Lichao Sun. Splitfed: When federated learning meets split learning, 2022.

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[46] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data, 2018.

[47] Praneeth Vepakomma, Abhishek Singh, Otkrist Gupta, and Ramesh Raskar. Nopeek: Information leakage reduction to share activations in distributed deep learning. In *2020 International Conference on Data Mining Workshops (ICDMW)*, pages 933–942, 2020.

[48] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity, 2020.

[49] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7794–7803, 2018.

[50] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[51] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[52] Song Xia, Yi Yu, Wenhan Yang, Meiwen Ding, Zhuo Chen, Ling-Yu Duan, Alex C. Kot, and Xudong Jiang. Theoretical insights in model inversion robustness and conditional entropy maximization for collaborative inference systems, 2025.

[53] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems (NeurIPS 2020)*, 2020.

[54] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.