Excellent idea! Clearly outlining the formulas, especially highlighting any novel aspects or novel applications, is crucial for an IEEE paper. Let's break them down.

Key Formulae and Their Roles in FEDMED:

Here's a brief on the main mathematical or procedural concepts used in FEDMED:

1. **Standard Federated Averaging (FedAvg) - Baseline Aggregation:**
   - **Client Update (Model Delta):** `Δw_k = w_k_local - w_global`
     - `w_global`: Global model parameters at the start of the round.
     - `w_k_local`: Local model parameters of client `k` after local training.
     - `Δw_k`: The change (delta) in model parameters for client `k`. Clients send this (or an encrypted version) to the server.
     - **Role:** Standard way for clients to communicate their learning.
     - **Novelty:** Not novel in itself; it's the standard FedAvg update.
   - **Server Aggregation (Std. FedAvg):** `w_global_new = w_global + Σ (n_k / N) * Δw_k_decrypted` (summation over selected clients `k`)
     - `n_k`: Number of data samples on client `k`.
     - `N`: Total number of data samples across all selected clients (`Σ n_k`).
     - `Δw_k_decrypted`: Decrypted model delta from client `k`.
     - **Role:** Baseline aggregation strategy. Weights updates based on the amount of data a client has.
     - **Novelty:** Not novel; this is the canonical FedAvg aggregation. You implemented this in Step 3 and 8a.

2. **Homomorphic Encryption (TenSEAL CKKS) - Privacy:**
   - **Client-Side Encryption:** `Enc_Δw_k = Encrypt_CKKS(Δw_k, pk_server)`
     - `Δw_k`: Plaintext model delta (numpy vector of floats).
     - `pk_server`: TenSEAL context (containing public key components) provided by the server.
     - `Encrypt_CKKS`: The TenSEAL CKKS encryption function (`ts.ckks_vector(context, data)`).
     - `Enc_Δw_k`: Encrypted model delta (serialized CKKSVector).
     - **Role:** Protects the confidentiality of individual client model updates from the server and other clients.
     - **Novelty:** Application of HE (specifically CKKS via TenSEAL) to FL model updates is an established research area. Your implementation is a practical realization. The novelty isn't the HE scheme itself, but its integration within the FEDMED framework.
   - **Server-Side Homomorphic Operations & Decryption (during aggregation):**
     - Weighted sum: `Σ (weight_k * Enc_Δw_k)` can be partially done homomorphically. `Agg_Enc_Δw = Σ_k ( Enc_Δw_k * plain_weight_k )` (CKKS: EncryptedVector * PlainScalar)

    - Decryption: `Agg_Δw_decrypted = Decrypt_CKKS(Agg_Enc_Δw, sk_server)`
        - `plain_weight_k`: The aggregation weight for client `k` (calculated by the server).
        - `sk_server`: TenSEAL context containing the server's secret key.
        - `Decrypt_CKKS`: The TenSEAL CKKS decryption function (`vector.decrypt()`).
    - **Role:** Allows the server to compute the weighted sum of encrypted deltas without seeing individual deltas, then decrypt only the final aggregate.
    - **Novelty:** Standard HE operations within a privacy-preserving aggregation scheme.

3. FEDMED's Privacy-Preserving Quality Score - Quality Assessment:

- **Formula:** `QS_k = 1 / (L_k +  )`
    - `QS_k`: Quality score for client `k`.
    - `L_k`: Average local training loss of client `k` over its local epochs.
    - (epsilon, e.g., `1e-6`): A small constant to prevent division by zero and to bound the score if loss is very low.
    - **Role:** Provides a scalar value representing the quality of client `k`'s local training. Lower loss (better training) results in a higher quality score. Sent in plaintext by the client.
    - **Novelty:**
        - Using inverse local loss as a quality metric is a known heuristic.
        - The **privacy-preserving aspect** is that it doesn't directly reveal client data or specific model parameters, only an aggregate performance statistic from local training.
        - Its **application within your specific FEDMED framework** as the primary driver for adaptive aggregation is part of your system's design.

4. FEDMED's Quality-Aware Aggregation (Core Mechanism):

- **Weight Calculation (based on Quality Scores):** `weight_k = Clipped_QS_k / Σ_j (Clipped_QS_j)`
    - `Clipped_QS_k`: The (potentially clipped) quality score of client `k`.
    - The summation `Σ_j (Clipped_QS_j)` is over all selected clients `j`.
    - `weight_k`: The final aggregation weight for client `k`'s model delta.
    - **Role:** To give higher influence to clients whose local training (as indicated by their reported quality score) is deemed of higher quality.
    - **Novelty:** This adaptive weighting based on the privacy-preserving quality score `QS_k` is a core component of FEDMED's novelty.
- **Server Aggregation (FEDMED):** `w_global_new = w_global + Σ_k (weight_k * Δw_k_decrypted)`
    - This looks like the FedAvg formula, but `weight_k` is now derived from quality scores instead of just sample sizes.

  - **Role:** Updates the global model using the adaptively determined weights.
  - **Novelty:** The adaptive aggregation strategy itself, driven by the quality scores.

5. FEDMED's Robust Score Aggregation - Defense against Dishonest Scores:

This is where you apply a mechanism to `QS_k` to get `Clipped_QS_k` before calculating the final `weight_k`.

- **a) Percentile-based Clipping (Evaluated in Step 1/6a, Step 4a):** `Lower_Bound = Percentile(all_QS, p_lower)` `Upper_Bound = Percentile(all_QS, p_upper)` `Clipped_QS_k = clip(QS_k, Lower_Bound, Upper_Bound)`
  - `all_QS`: The set of reported quality scores from all participating clients in a round.
  - `p_lower`, `p_upper`: Lower and upper percentile thresholds (e.g., 5th and 95th).
  - `clip(value, min, max)`: Standard clipping function.
  - **Role:** To limit the influence of extreme outlier quality scores.
  - **Novelty:** Applying percentile clipping to client-reported quality scores in FL is a reasonable heuristic for robustness. Its limitations were shown in your Step 4a.
- **b) MAD-based Clipping (Implemented in Step 5, Step 6b - KEY NOVELTY):**
  1. `Median_QS = median(all_QS)`
  2. `MAD_scaled = median_abs_deviation(all_QS, scale='normal')` (approximates standard deviation if data were normal: `median(|QS_i - Median_QS|) / 0.6745`)
  3. `K_MAD = 3.0` (a configurable sensitivity parameter, typically 2 or 3)
  4. `Lower_Bound_MAD = Median_QS - K_MAD * MAD_scaled`
  5. `Upper_Bound_MAD = Median_QS + K_MAD * MAD_scaled`
  6. `Lower_Bound = max(0.0, Lower_Bound_MAD)` (Ensuring non-negativity for scores)
  7. `Upper_Bound = Upper_Bound_MAD`
  8. `Clipped_QS_k = clip(QS_k, Lower_Bound, Upper_Bound)`
  - **Role:** To provide a more robust method (compared to simple percentiles when faced with colluding attackers) for identifying and clipping outlier quality scores, especially those dishonestly inflated.
  - **Novelty:**
    - The **application of MAD-based clipping specifically to client-reported quality scores in a privacy-preserving FL framework** to defend against dishonest score reporting is a significant part of FEDMED's novel contribution.
    - While MAD is a known robust statistic, its use in this context to sanitize quality scores before they influence HE-protected aggre-

gation is the novel aspect. Your experiments (Step 5 vs. 4a/4b) demonstrated its effectiveness.

Summary of Novelty in FEDMED's Formulae/Mechanisms:

1. **The Overall System Design:** The combination of:
   - HE for update privacy (TenSEAL CKKS).
   - A specific privacy-preserving quality score (1/loss).
   - Adaptive aggregation based on these scores.
   - **Crucially, the MAD-based robust aggregation of these quality scores as a defense against score manipulation.**

2. **Privacy-Preserving Quality Score in Context:** While 1/loss isn't new as a metric, its use as a *plaintext scalar reported alongside HE-encrypted updates* to drive adaptive aggregation is a specific design choice within your privacy-preserving framework.

3. **MAD-based Robust Score Aggregation:** This is the strongest novel component demonstrated by your experiments. It directly addresses a vulnerability in simpler quality-aware systems. You showed it works where percentile clipping failed against certain attacks.

When writing your paper, you'd introduce each concept, define it (with formula if applicable), state its role, and then, for the components you are proposing as novel or a novel combination/application (like the MAD-based score clipping), you would emphasize that. The experimental results then serve to validate the effectiveness of these novel aspects.