GaussianMatch: Semi-Supervised Regression with Pseudo-Label Filtering via Multi-View Gaussian Consistency

Supplementary Material

A. Theoretical Proofs Related to Gaussian Consistency Filter (GCF)

A.1. Proof of $\rho = \sqrt{-2\ln\tau}$

We derive the equivalence between the Gaussian similarity threshold and the standard deviation-based filtering criterion.

Given a set of K predictions $\{y_j^{(k)}\}_{k=1}^K$ for an unlabeled sample $x_j^{(u)}$, we define:

$$\mu_j = \frac{1}{K}\sum_{k=1}^{K} y_j^{(k)} \tag{1}$$

$$\sigma_j = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(y_j^{(k)} - \mu_j)^2} \tag{2}$$

The Gaussian similarity score for each view is:

$$S_j(k) = \exp\left(-\frac{(y_j^{(k)} - \mu_j)^2}{2\sigma_j^2}\right) \tag{3}$$

To retain a pseudo-label, the minimum similarity across all views must exceed a threshold τ:

$$\min_{1\le k\le K} S_j(k) \ge \tau \tag{4}$$

Substituting the expression for $S_j(k)$, we get:

$$\exp\left(-\frac{(y_j^{(k)} - \mu_j)^2}{2\sigma_j^2}\right) \ge \tau \tag{5}$$

$$-\frac{(y_j^{(k)} - \mu_j)^2}{2\sigma_j^2} \ge \ln\tau \tag{6}$$

$$(y_j^{(k)} - \mu_j)^2 \le -2\sigma_j^2\ln\tau \tag{7}$$

$$|y_j^{(k)} - \mu_j| \le \sigma_j\sqrt{-2\ln\tau} \tag{8}$$

Define:

$$\rho = \sqrt{-2\ln\tau} \tag{9}$$

Then the filtering criterion becomes:

$$|y_j^{(k)} - \mu_j| \le \rho\sigma_j, \quad \forall k \in 1,\dots,K \tag{10}$$

Thus, retaining a pseudo-label based on Gaussian similarity τ is equivalent to requiring all predictions to lie within ρ standard deviations of their mean.

A.2. Theoretical Justification for Gaussian Consistency Scoring

To address the lack of formal guarantees behind the Gaussian consistency score, we present the following theoretical result linking prediction consistency and pseudo-label reliability.

Theorem 1 (Consistency-Confidence Duality)**.** *Let $f : \mathcal{X} \to \Delta^C$ be a model mapping an input to the probability simplex over C classes. Given k weakly augmented views $\{\tilde{x}_i\}_{i=1}^k$ of a sample x, let $p_i = f(\tilde{x}_i)$ be their corresponding predictions. Define:*

- ***Prediction mean:*** $\mu_p = \frac{1}{k}\sum_{i=1}^k p_i$
- ***Consistency variance:*** $\sigma_p^2 = \frac{1}{k}\sum_{i=1}^k \|p_i - \mu_p\|^2$
- ***Gaussian similarity:*** $s_{ij} = \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma^2}\right)$

Then, for any $\epsilon > 0$, the pseudo-label reliability is bounded by:

$$\mathbb{P}\left(\|\mu_p - y^*\|_1 \le \epsilon\right) \ge 1 - \frac{\sigma_p^2}{k\epsilon^2} - \delta(\sigma) \tag{11}$$

where y^* is the true label and $\delta(\sigma) \to 0$ as $\sigma \to \infty$.

Proof. **1. Model robustness implies low variance:** Assume model f is ϵ-robust at input x, i.e., $\exists\mu \in \mathbb{R}^C$ such that $\|f(\tilde{x}_i) - \mu\| \le \epsilon$ for all i. Then, by the triangle inequality:

$$\sigma_p^2 = \frac{1}{k}\sum_{i=1}^k \|p_i - \mu_p\|^2 \le \epsilon^2 \tag{12}$$

2. Variance controls pseudo-label error: Applying Chebyshev's inequality:

$$\mathbb{P}\left(\|\mu_p - \mathbb{E}[\mu_p]\| \ge \eta\right) \le \frac{\sigma_p^2}{k\eta^2} \tag{13}$$

If the model is calibrated such that $\|\mathbb{E}[\mu_p] - y^*\| \le \gamma$, then:

$$\mathbb{P}\left(\|\mu_p - y^*\| \ge \eta + \gamma\right) \le \frac{\sigma_p^2}{k\eta^2} \tag{14}$$

3. Gaussian similarity reflects variance: By Jensen's inequality:

$$\mathbb{E}[s_{ij}] = \mathbb{E}\left[\exp\left(-\tfrac{\|p_i - p_j\|^2}{2\sigma^2}\right)\right] \le \exp\left(-\tfrac{1}{2\sigma^2}\left(\mathbb{E}[\|p_i - p_j\|]\right)^2\right) \tag{15}$$

When $\sigma_p^2 > \theta$, the expected pairwise deviation $\mathbb{E}[\|p_i - p_j\|] \ge \sqrt{2\theta}$, implying $\mathbb{E}[s_{ij}] \to 0$. □

Corollary 1 (Formal Guarantee for Pseudo-Label Filtering)**.** *Let the mean similarity score be:*

$$\bar{s} = \frac{2}{k(k-1)} \sum_{i<j} s_{ij} \tag{16}$$

Then, setting a similarity threshold τ yields:

$$\mathbb{P}(\|\mu_p - y^*\| \geq \epsilon) \leq \frac{1}{\tau k \epsilon^2} + C(\sigma, k) \tag{17}$$

where

$$C(\sigma, k) = \underbrace{e^{-\mathcal{O}(k\sigma^2)}}_{\textit{Gaussian tail}} + \underbrace{\|\mathbb{E}[\mu_p] - y^*\|}_{\textit{Bias term}} \tag{18}$$

and $C(\sigma, k) \to 0$ as $k \to \infty$ or $\sigma \to 0$.

Interpretation:

- Low variance σ_p^2 (i.e., high similarity $\bar{s}$) implies that the pseudo-label mean μ_p is concentrated near the true label y^*.
- High variance (low $\bar{s}$) indicates unreliable predictions and higher pseudo-label error.
- Parameters: σ tunes sensitivity to prediction variance; τ sets the filtering threshold to control reliability.

B. Pseudocode for Gaussian Consistency Filtering

To enhance pseudo-label quality in semi-supervised regression, we introduce the Gaussian Consistency Filter (GCF), a principled and adaptive mechanism that evaluates the reliability of unlabeled predictions based on statistical agreement across multiple augmented views. As illustrated in Algorithm 1, GCF computes the mean and variance of predictions obtained from weak augmentations, applies a Bayesian smoothing strategy to stabilize early stage estimation, and evaluates similarity scores via a Gaussian kernel. A pseudo label is retained only if all similarity scores exceed a confidence threshold, effectively filtering out inconsistent or noisy samples. This procedure encourages the model to focus on high-confidence regions in the data distribution and plays a crucial role in mitigating error propagation during training.

Algorithm 1 Gaussian Consistency Filter with Adaptive Smoothing

Require:
- Unlabeled sample: u_j
- Weak augmentation function: $\alpha(\cdot)$
- Current iteration: t
- Total iterations: t_{total}
- Bayesian parameters: $\alpha_0 \geq 1, \beta_{\min} > 0$

Ensure:
- Pseudo-label mask: $\tilde{\mathcal{M}}(u_j) \in \{0, 1\}$
- Consensus prediction: μ_j

1. Generate Augmented Views and Predictions:

for $k = 1$ **to** K **do**
  $\tilde{u}_j^{(k)} \leftarrow \alpha(u_j)$ // Generate weakly-augmented sample
  $q_j^{(k)} \leftarrow \text{Model}(\tilde{u}_j^{(k)})$ // Model prediction
end for

2. Compute Statistical Measures:

$\mu_j \leftarrow \frac{1}{K} \sum_{k=1}^{K} q_j^{(k)}$ // Compute mean prediction
$\sigma_j \leftarrow \sqrt{\frac{1}{K} \sum_{k=1}^{K} (q_j^{(k)} - \mu_j)^2}$ // Raw standard deviation

3. Adaptive Variance Smoothing:

if $t \leq t_w$ **then**
  $\beta_t \leftarrow \bar{\beta}_w$ // Warmup phase: initial mean
else
  $\gamma(t) \leftarrow \frac{t - t_w}{t_{\text{total}} - t_w}$ // Decay coefficient
  $\beta_t \leftarrow \max\left(\bar{\beta}_w(1 - \gamma(t)), \beta_{\min}\right)$ // Dynamically adjust baseline variance
end if
$\hat{\sigma}_j \leftarrow \sqrt{\frac{\beta_t + \frac{1}{2} \sum_{k=1}^{K} (q_j^{(k)} - \mu_j)^2}{\alpha_0 + \frac{K}{2} - 1}}$ // Bayesian - smoothed std dev

4. Similarity Evaluation:

for $k = 1$ **to** K **do**
  $S_j(k) \leftarrow \exp\left(-\frac{(q_j^{(k)} - \mu_j)^2}{2\hat{\sigma}_j^2}\right)$ // Gaussian similarity score
end for

5. Pseudo-Label Mask:

if $\min_{1 \leq k \leq K} S_j(k) \geq \tau$ **then**
  $\tilde{\mathcal{M}}(u_j) \leftarrow 1$ // Retain valid pseudo-label
else
  $\tilde{\mathcal{M}}(u_j) \leftarrow 0$ // Filter noisy sample
end if
return $\tilde{\mathcal{M}}(u_j), \mu_j$ // Return mask and reliable pseudo-label

C. More Dataset Details

UTKFace [8] is a large-scale facial dataset that, while initially designed for age, gender, and ethnicity estimation, has proven valuable for semi-supervised regression tasks. It comprises 23,708 images spanning a broad range of ages, ethnicities, and facial orientations, making it an ideal benchmark for evaluating regression models under diverse demographic conditions. The inherent variations in lighting, pose, and background introduce realistic challenges that closely simulate real-world scenarios, while the rich and detailed annotations facilitate the investigation of continuous attributes. Overall, UTKFace provides a robust foundation for developing and testing semi-supervised regression methodologies in multimedia analysis.

VIPL [4], the data set contains 107 subjects, and each

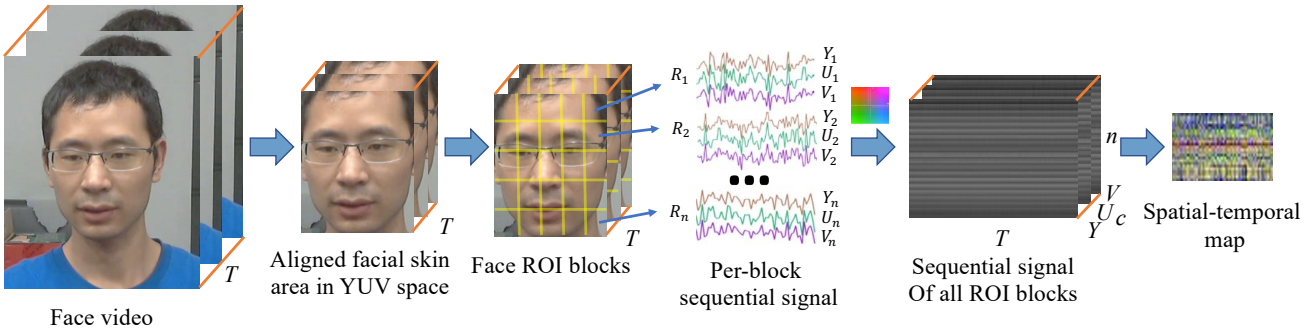

Figure 1. An overview of the process: We first align faces across frames using detected landmarks [6] and convert the aligned images to YUV color space. Next, the facial region is divided into n ROI blocks, and the average color for each channel is computed in each block. The averages for the same block from different frames are concatenated (e.g., $Y_1, U_1, V_1, Y_2, U_2, V_2, \ldots, Y_n, U_n, V_n$) and arranged as rows to form a $T \times n \times c$ spatial-temporal map.

object takes videos of 9 scenarios, including v1 (stable scenario), v2 (motion scenario), v3 (talking scenario), v4 (dark scenario), v5 (bright scenario), v6 (long distance scenario), v7 (exercise scenario), v8 (phone stable scenario), v9 (phone motion scenario). Nine scenarios were collected with four different acquisition devices, named as source1-4. Notably, this paper does not study the NIR data sources, and only sources1-3 are used. Besides, in this paper, we do not use the uncalibrated BVP signal of the dataset for the time being and only use the HR value labels. Additionally, we segment the videos into clips of 256 frames using a sliding window with a step size of 50 frames. In the training set, we generated 268,162 clips, while 66,398 clips were produced for the test set. During training, following [3, 5], we convert these clips into STMaps as the model's input, and the detailed process for generating spatial-temporal maps (STMaps) is shown in Figure 1. Each STMap captures temporal variations in color and motion across facial regions, effectively summarizing physiological dynamics. These maps are structured as image-like representations that preserve both spatial and temporal information. As a result, they are directly compatible with standard ResNet backbones without requiring any architectural modifications.

Yelp Review [7], The Yelp Review dataset is widely used for sentiment and opinion mining, where the task is to predict user ratings based on textual reviews. Each review is associated with a rating from 0 to 4, corresponding to increasing levels of satisfaction. In our experiments, we adopt the preprocessed version of the dataset provided by the USB benchmark, which includes 250,000 samples for training, 25,000 for validation, and 10,000 for testing. For our setting, only the training and validation splits are used for model development and evaluation, respectively.

D. More Implementation Details.

To ensure a fair and consistent comparison across all baseline methods, we adopt unified training protocols, including optimizer choices, learning rate schedules, and batch sizes, unless otherwise specified. For all experiments, we use the Mean Absolute Error (MAE) as the loss function and perform evaluations at fixed intervals to maintain stability across different input modalities. Hyperparameters such as weight decay, learning rate, and layer decay are tuned individually to accommodate the characteristics of each modality while keeping the overall training strategy aligned. All experiments are conducted on a machine running Ubuntu 22.04.5 LTS with dual AMD EPYC 7T83 64-core processors (256 threads), 503 GB of RAM, and an NVIDIA GeForce RTX 4090 GPU with 24 GB memory.

GaussianMatch employs a variance smoothing parameter β, which is gradually introduced through a warmup phase of 3,000 iterations to stabilize the early stages of training. The unsupervised loss is weighted by $\lambda_u = 15$, which balances the contributions from labeled and unlabeled branches. Additionally, we incorporate the MixUp augmentation strategy from MixMatch to enhance data diversity, which is a critical factor in semi-supervised learning. Unlike the original approach, we extend MixUp to include not only weakly augmented labeled and unlabeled data but also strongly augmented unlabeled samples. This enables the model to capture invariant representations while promoting robustness through consistency training.

For modality-specific regression tasks, we adopt Wide ResNet-28-2 for facial image-based inputs, RhythmNet for heart rate estimation from face videos, and BERT-Small for textual data. These architectures are selected based on their strong performance in prior work and their suitability for each data type. As summarized in Table 1, each model is configured with modality-aware hyperparameters such as

Table 1. Training configuration details for Wide ResNet-28-2, RhythmNet [5], and BERT-Small.

Configuration	Wide ResNet-28-2	RhythmNet	Bert-Small
Training Iterations	262,144	102,400	102,400
Evaluation Iterations	1,024	1,024	1,024
Training Batch Size	32	4	8
Optimizer	SGD	Adam	AdamW
Momentum	0.9	-	-
Criterion	MAE	MAE	MAE
Weight Decay	1e-03	1e-04	5e-04
Layer Decay	1.0	0.75	0.75
Learning Rate	1e-02	1e-03	1e-05
EMA Weight	0.999	-	-
Pretrained	False	False	True
Sampler	Random	Random	Random
Grids Size	-	5x5	-
Image Resize	40x40	25x300	-
Temporal Length	-	300	-
Max Length	-	-	512

input resolution, maximum sequence length, and training iterations. Pretrained weights are used for BERT-Small, while vision and video models are trained from scratch to better evaluate their semi-supervised learning capabilities.

E. Compared with Uncertainty-Based SSR

SimRegMatch [2] improves pseudo-label quality in semi-supervised regression by combining uncertainty-based filtering with similarity-based calibration. To provide a meaningful comparison, we evaluate SimRegMatch under the same UTKFace setting with 250 labeled samples. As shown in Table 2, our method outperforms SimRegMatch on all metrics, demonstrating its effectiveness and strong performance in semi-supervised regression.

Table 2. Comparison with SimRegMatch on UTKFace (250 labels)

Method	MAE↓	R^2↑	SRCC↑
Supervised	9.42	0.540	0.712
RankUp	7.06	0.751	0.835
SimRegMatch	7.84	0.702	0.809
GaussianMatch (Ours)	**6.38**	**0.794**	**0.862**

F. Additional Visualizations

F.1. Feature Space Analysis via t-SNE

Figure 2 presents a t-SNE visualization of feature embeddings learned by eight different methods on the UTKFace dataset: FullySupervised, Supervised, CLSS, Mean Teacher, Pi-Model, MixMatch, RankUp, and GaussianMatch. The FullySupervised model, trained with complete label supervision, demonstrates an ideal embedding structure, where feature clusters exhibit smooth transitions between semantically adjacent age groups (e.g., between the 0–25 and 25–50 year ranges). This structure reflects high-quality semantic representations.

In contrast, the Supervised baseline, trained with only limited labeled data, produces fragmented clusters and irregular transitions, indicating poor generalization and insufficient feature discrimination. Similarly, CLSS, Mean Teacher, Π-Model, and MixMatch exhibit fragmented and entangled feature spaces, leading to degraded performance. RankUp achieves moderate improvement, with compact clusters observed primarily in densely labeled age ranges (e.g., 25–50 years). However, it struggles significantly in label-sparse regions such as very young (0–5 years) and very old (75–100 years) groups. This is evident from the irregular and disconnected clusters, such as scattered points near coordinates $(-50, 25)$ representing the 0–5 year group. These artifacts suggest that RankUp is sensitive to under-represented samples and lacks robustness in extrapolating

to rare age values.

In comparison, GaussianMatch exhibits a more coherent and continuous embedding structure across the entire age spectrum. Even in the extreme age ranges (e.g., 0–5 years near (−10, 15) and 75–100 years around (30,−20)), the feature clusters remain compact and well-aligned. This robustness stems from the proposed Gaussian Consistency Filter, which actively filters out pseudo-labels with high predictive variance. By retaining only reliable, consensus-driven pseudo-labels, GaussianMatch avoids noise accumulation and learns high-quality features. Notably, its learned embedding is structurally closer to that of the FullySupervised model, reflecting improved generalization and pseudo-label fidelity even in highly underrepresented regions.

F.2. Training Dynamics and Representation

To visualize the noise suppression mechanism, we tracked 1,000 samples with high initial errors during the training process, as illustrated in Figure 3. Relaxed early filtering helps the model bootstrap representation learning, while progressively stricter filtering effectively blocks gradients from noisy samples as training proceeds. Ultimately, the masking rate converges to ∼14% rather than zero. This maintains a safety barrier for ambiguous cases and empirically confirms that our method functions as an effective noise suppression mechanism.

G. More Ablation Study

We analyze parameter sensitivity in Table 3 by varying one parameter at a time and reporting MAE on UTKFace with 250 labeled samples. The low variance of MAE for β_0 (0.027) and α_0 (0.007) shows the model is insensitive to them, reflecting the stability of our Bayesian smoothing. In contrast, τ has a higher variance (0.131). Excessively low τ values (0.60 or 0.70) weaken filtering, letting noisy pseudo-labels through and harming performance. This supports choosing τ between 0.90 and 0.95 for strong consistency and better pseudo-labels.

Table 3. Ablation study on the sensitivity of key hyperparameters: τ (confidence threshold), β_0 (EMA smoothing strength), and α_0 (smoothing sharpness). Default settings are in bold.

Parameter	Setting	MAE↓	R^2↑	SRCC↑
τ	**0.95 (default)**	**6.38**±0.06	**0.794**±0.005	**0.862**±0.007
	0.90	6.58±0.06	0.777±0.007	0.850±0.007
	0.80	6.97±0.07	0.753±0.008	0.841±0.009
	0.70	7.03±0.10	0.730±0.010	0.833±0.012
	0.60	7.36±0.12	0.738±0.013	0.832±0.013
β_0	1e-2	6.54±0.08	0.784±0.009	0.851±0.010
	2e-2 (default)	**6.38**±0.06	**0.794**±0.005	**0.862**±0.007
	5e-2	6.68±0.07	0.776±0.006	0.847±0.006
	1e-1	6.68±0.09	0.779±0.010	0.845±0.010
	5e-1	6.81±0.06	0.765±0.006	0.840±0.007
α_0	1	6.45±0.06	0.780±0.006	0.860±0.006
	2 (default)	**6.38**±0.06	**0.794**±0.005	**0.862**±0.007
	3	6.49±0.05	0.790±0.005	0.857±0.005
	5	6.50±0.07	0.778±0.006	0.851±0.006
	10	6.60±0.07	0.781±0.008	0.852±0.008

We provide a runtime analysis in Table 4, showing that even with $K = 8$, which gives the best MAE (6.38), the per-batch cost is only 121 ms. Smaller K values further reduce cost with minimal performance drop, demonstrating our method's efficiency and controllable resource usage.

K	Train(ms)	PL(ms)	Bayes(ms)	Mean/Std(ms)	MAE↓
2	65	7	∼0	∼0	6.74 ± 0.04
4	92	13	∼0	∼0	6.48 ± 0.07
8	121	25	∼0	∼0	6.38 ± 0.06
16	168	52	∼0	∼0	6.52 ± 0.06

Table 4. Effect of K on runtime and MAE. Train denotes the total training time, Bayes refers to Bayesian smoothing, and PL represents pseudo-label generation and statistical computations (mean/std). These components contribute minimally to the overall runtime.

To validate our choice of Gaussian modeling over other consistency measures, we evaluated several alternative filtering strategies. We compared GaussianMatch against three alternative criteria: Variance Thresholding, Quantile Filtering, and Median Aggregation. As shown in Table 5, Gaussian modeling yields a clear empirical gain. GaussianMatch achieves the lowest MAE (6.38) on UTKFace with 250 labels, outperforming Variance Thresholding (7.11), Quantile Filtering (6.92), and Median Aggregation (6.96). This confirms that modeling prediction consistency as a normal distribution provides a flexible and continuous evaluation, making it more suitable for regression tasks than other methods.

Table 5. Comparison with alternative filtering strategies.

Method	Var. Thresh.	Quantile Filter	Median Agg.	GaussianMatch (Ours)
MAE (UTKFace 250 labels)	7.11	6.92	6.96	**6.38**

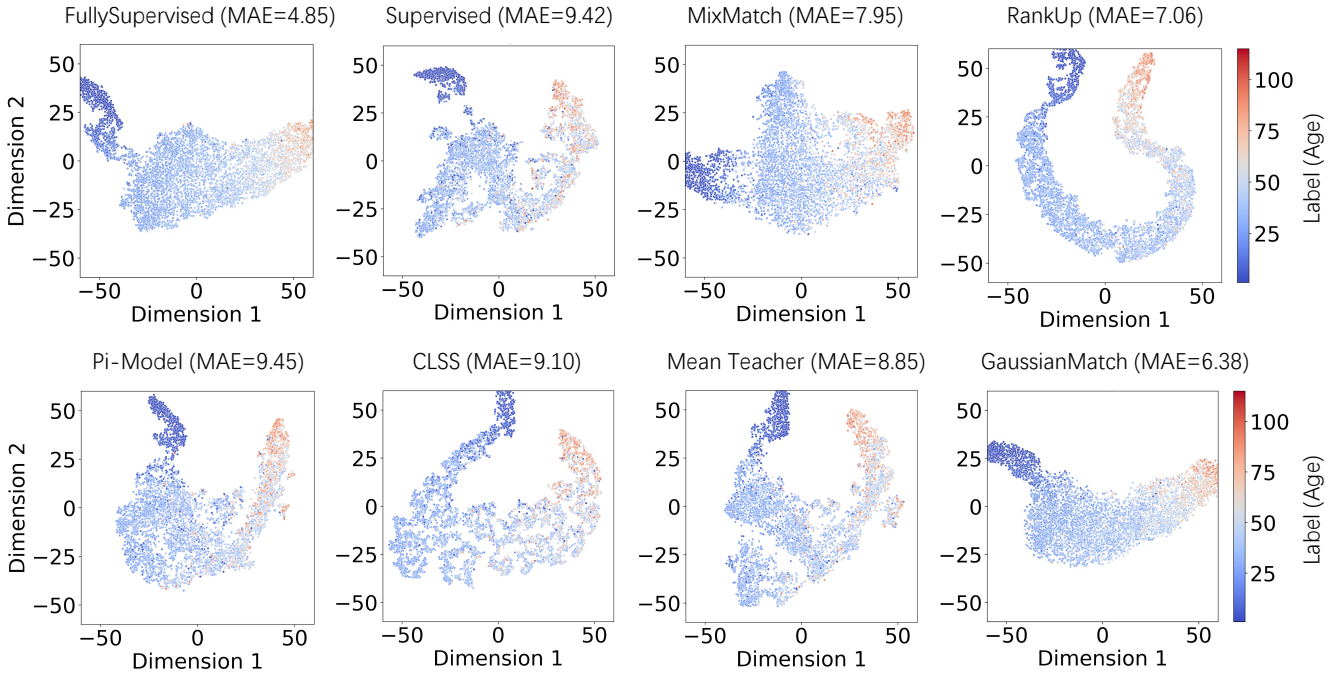

Figure 2. t-SNE visualization of feature spaces across methods.

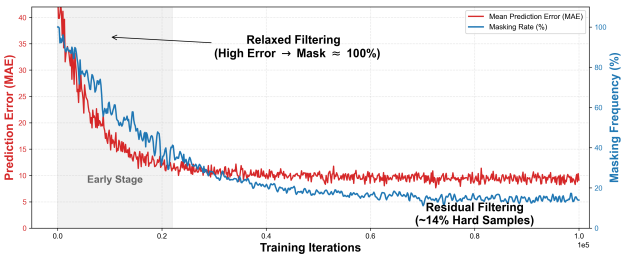

Figure 3. The noise suppression process for high-error samples.

H. Notation and Symbols

In this section, we summarize and clarify the notations and symbols (Table 6) used throughout the paper. This provides readers with a convenient reference to understand the mathematical expressions and terminologies employed in the methodology and experimental descriptions.

H.1. How GaussianMatch Differs from UCVME and RankUp?

We summarize the key differences among UCVME, RankUp, and our method GaussianMatch in Table 7. UCVME adopts a dropout-based uncertainty estimation mechanism but does not explicitly filter pseudo-labels, resulting in unstable predictions. RankUp focuses on ranking consistency but lacks mechanisms to control or filter the label values, leading to potential noise accumulation. In contrast, GaussianMatch introduces a Gaussian-based label filtering mechanism that explicitly uses multi-view prediction agreement to improve pseudo-label reliability. This enables the method to suppress noise and maintain high-quality supervision in the regression setting.

H.2. Why not test classical SSC methods?

We did not include classical SSC methods such as FreeMatch, FlatMatch [1], SoftMatch, and others in our comparison because they fundamentally rely on probability-based thresholding over discrete class predictions. These methods are inherently designed for classification tasks and their core mechanisms break down when the classification head is replaced with a regression output, making them unsuitable for SSR. In contrast, MixMatch does not depend on class probabilities and can be adapted to SSR with minimal modifications while preserving its underlying framework. Therefore, we include MixMatch as a baseline but exclude the more recent SSC methods to ensure a fair and meaningful comparison.

Table 6. Notation Summary

Symbol	Meaning
$\mathcal{D}_L$	Labeled dataset with input features and continuous regression targets
$\mathcal{D}_U$	Unlabeled dataset with input features only
$x_i, u_j \in \mathbb{R}^d$	Input feature vectors of dimension d
$y_i \in \mathbb{R}$	Continuous regression target
$f_\theta : \mathbb{R}^d \to \mathbb{R}$	Regression model parameterized by θ
$\lvert\mathcal{X}\rvert, \lvert\mathcal{U}\rvert$	Batch size of labeled and unlabeled data
$\mathcal{X}', \mathcal{U}'$	Mixed labeled and unlabeled batches from MixUp
$\mathcal{R}(\mathcal{D}_U; \theta)$	Unsupervised regularization term on unlabeled data
λ	Weight for regularization term
K	Number of weakly augmented views per unlabeled sample
$\tilde{u}_j^{(k)}$	The k-th weakly augmented sample
$q_j^{(k)} = f_\theta(\tilde{u}_j^{(k)})$	Model prediction for the k-th augmented view
μ_j	Consensus value (mean prediction) across augmentations
σ_j	Standard deviation of predictions
$\hat{\sigma}_j$	Smoothed standard deviation to avoid over-strict filtering
ρ	Confidence interval multiplier derived from threshold τ
$S_j(k)$	Gaussian similarity score measuring agreement with consensus
$\tilde{\mathcal{M}}(u_j)$	Pseudo-label mask indicating reliable samples
β_0, α_0	Bayesian smoothing parameters controlling variance regularization
β_t	Adaptive smoothing parameter decayed during training
$\alpha(\cdot)$, $\mathcal{A}(\cdot)$	weakly and strongly augmentation
$H(\cdot, \cdot)$	cross-entropy loss
$\bar{q}$	mean prediction from K weak augmentations as pseudo-labels for regression
p, q	sharpened pseudo-label distributions for labeled and unlabeled data
λ_u	Weight for unsupervised consistency loss

Table 7. Comparison of recent semi-supervised regression (SSR) methods

Method	Core Idea	Confidence Handling	Consistency Strategy	Pseudo-label Quality
UCVME	Dropout-based uncertainty estimation	Implicit, via variance	Variational inference	Unstable due to lack of filtering
RankUp	Pairwise ranking supervision	Explicit, ranking classifier	Ranking consistency	Uncontrolled label values, may propagate noise
GaussianMatch	Gaussian-based label filtering	Explicit, Gaussian similarity	Multi-view Gaussian consistency	High-quality labels with noise suppression via consistency filtering

References

[1] Zhuo Huang, Li Shen, Jun Yu, Bo Han, and Tongliang Liu. Flatmatch: Bridging labeled data and unlabeled data with cross-sharpness for semi-supervised learning. *Advances in neural information processing systems*, 36:18474–18494, 2023.

[2] Yongwon Jo, Hyungu Kahng, and Seoung Bum Kim. Deep semi-supervised regression via pseudo-label filtering and calibration. *Applied Soft Computing*, 161:111670, 2024.

[3] Hao Lu, Zitong Yu, Xuesong Niu, and Ying-Cong Chen. Neuron structure modeling for generalizable remote physiological measurement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18589–18599, 2023.

[4] Xuesong Niu, Hu Han, Shiguang Shan, and Xilin Chen. Viplhr: A multi-modal database for pulse estimation from less-constrained face video. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part V 14*, pages 562–576. Springer, 2019.

[5] Xuesong Niu, Shiguang Shan, Hu Han, and Xilin Chen. Rhythmnet: End-to-end heart rate estimation from face via spatial-temporal representation. *IEEE Transactions on Image Processing*, 29:2409–2423, 2019.

[6] SeetaFace. Seetaface: A high-performance face recognition engine, 2014. Available: `http://www.seetaface.org`.

[7] Yidong Wang, Hao Chen, Yue Fan, Wang Sun, Ran Tao, Wenxin Hou, Renjie Wang, Linyi Yang, Zhi Zhou, Lan-Zhe Guo, et al. Usb: A unified semi-supervised learning benchmark for classification. *Advances in Neural Information Processing Systems*, 35:3938–3961, 2022.

[8] Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5810–5818, 2017.