1 Standard Contrastive Representation Distillation (CRD) (From the original paper)

Standard CRD operates in a self-supervised manner. Its goal is to make a student network's feature representation for a given sample mimic the teacher's representation for that **same** sample, while being distinct from the teacher's representations for all **other** samples.

1.1 Components and Dynamics

The setup for a single sample i (the "anchor") is as follows:

- **Anchor ($\mathbf{s}_i$):** The feature vector from the **student network** for sample i. **This is the only component that moves**; it receives all gradients.
- **Positive ($\mathbf{t}_i$):** The feature vector from the **teacher network** for the *exact same sample* i. This is a **fixed target** in the embedding space.
- **Negatives ($\{\mathbf{t}_j\}_{j \neq i}$):** The set of feature vectors from the **teacher network** for all *other samples* j in the batch or memory bank. These are **fixed repellents**.

The loss function, a form of InfoNCE, enforces two dynamics:

1. **The "Pull":** The loss pulls the student anchor $\mathbf{s}_i$ closer to its corresponding teacher positive $\mathbf{t}_i$.
2. **The "Push":** The loss simultaneously pushes the student anchor $\mathbf{s}_i$ away from *all* teacher negatives $\{\mathbf{t}_j\}$.

1.2 The Loss Function

For a single student anchor $\mathbf{s}_i$, the loss is:

$$\mathcal{L}_{\text{CRD},i} = -\log\left(\frac{\exp(\text{sim}(\mathbf{s}_i, \mathbf{t}_i)/\tau)}{\exp(\text{sim}(\mathbf{s}_i, \mathbf{t}_i)/\tau) + \sum_{j \neq i} \exp(\text{sim}(\mathbf{s}_i, \mathbf{t}_j)/\tau)}\right)$$

where $\text{sim}(\cdot, \cdot)$ is a similarity function (e.g., dot product) and τ is a temperature hyperparameter. The student's parameters are updated to minimize this loss, moving $\mathbf{s}_i$ to "hit the target" $\mathbf{t}_i$ while dodging all "repellents" $\mathbf{t}_j$.

2 From Self-Supervised to Supervised Contrastive Distillation (SupCRD)

Supervised Contrastive Representation Distillation (SupCRD) incorporates label information while shaping the student's latent.

2.1 Components and Dynamics

The setup for a student anchor $\mathbf{s}_i$ (with label y_i) is:

- **Anchor ($\mathbf{s}_i$):** Same as before (student vector, y_i).
- **Positives ($P(i)$):** The set of *all* teacher vectors with the **same label** as the anchor. $P(i) = \{\mathbf{t}_k \mid y_k = y_i\}$.
- **Negatives ($N(i)$):** The set of *all* teacher vectors with **different labels**. $N(i) = \{\mathbf{t}_j \mid y_j \neq y_i\}$.

This modification directly structures the latent space based on class. The student's "dog" vector ($\mathbf{s}_i$) is now pulled towards the *entire cluster* of teacher "dog" vectors, and pushed away from *all* "cat", "truck", and "plane" vectors.

2.2 The Loss Function

The loss is modified to sum over all positives in the numerator:

$$\mathcal{L}_{\text{SupCRD},i} = -\log\left(\frac{\sum_{\mathbf{t}_k \in P(i)} \exp(\text{sim}(\mathbf{s}_i, \mathbf{t}_k)/\tau)}{\sum_{\mathbf{t}_k \in P(i)} \exp(\text{sim}(\mathbf{s}_i, \mathbf{t}_k)/\tau) + \sum_{\mathbf{t}_j \in N(i)} \exp(\text{sim}(\mathbf{s}_i, \mathbf{t}_j)/\tau)}\right)$$

This is a powerful loss, but it is still rigid. It pushes all negative *classes* away with equal force.

3 (MAIN PART) Proposed Method: Logit-Weighted SupCRD ($\sim$ Decoupled Feature Distillation)

We modify the SupCRD loss to distill the teacher's "dark knowledge." The core idea is to use the **teacher's output probabilities** for the *anchor sample* i to dynamically weight the push and pull forces.

Let $\mathbf{z}_i^T$ be the teacher's logit vector for the anchor sample i. Let $P_i^T = \text{softmax}(\mathbf{z}_i^T/T_{kd})$ be the teacher's soft probability distribution (where T_{kd} is the distillation temperature). Let $p_i^T(y_c)$ be the teacher's soft probability for any given class y_c.

3.1 The Logit-Weighted Loss Function

We introduce two new weighting terms, w_{pull} and w_{push}, into the SupCRD loss. These weights are controlled by base hyperparameters α and β (from DKD), and modulated by the teacher's probabilities to achieve **full decoupling** of target and non-target class knowledge.

- **Pull Weight (w_{pull}):** The "pull" force is now **adaptive**, modulated by the teacher's confidence on the target class. When the teacher is confident about the correct class (high $p_i^T(y_i)$), the pull is strengthened, as these are reliable examples that define the cluster center. When the teacher is uncertain (low $p_i^T(y_i)$), the pull is weakened, preventing hard/ambiguous examples from dominating the cluster structure.

$$w_{\text{pull},i} = \alpha \cdot p_i^T(y_i)$$

- **Push Weight (w_{push}):** The "push" force for each negative $\mathbf{t}_j$ (with class y_j) is scaled *inversely* to its semantic similarity. We use $(1 - p_i^T(y_j))$ as the weight, so that semantically distant negatives are pushed harder.

$$w_{\text{push},ij} = \beta \cdot (1 - p_i^T(y_j))$$

The new hybrid loss function for anchor $\mathbf{s}_i$ (with true label y_i) becomes:

$$\mathcal{L}_{\text{Hybrid},i} = -\log\left(\frac{(w_{\text{pull},i}) \sum_{\mathbf{t}_k \in P(i)} \exp(\text{sim}(\mathbf{s}_i, \mathbf{t}_k)/\tau)}{(w_{\text{pull},i}) \sum_{\mathbf{t}_k \in P(i)} \exp(\text{sim}(\mathbf{s}_i, \mathbf{t}_k)/\tau) + \sum_{\mathbf{t}_j \in N(i)} (w_{\text{push},ij}) \cdot \exp(\text{sim}(\mathbf{s}_i, \mathbf{t}_j)/\tau)}\right)$$

Substituting the definitions of the weights:

$$\mathcal{L}_{\text{Hybrid},i} = -\log\left(\frac{(\alpha \cdot p_i^T(y_i)) \sum_{\mathbf{t}_k \in P(i)} \exp(\text{sim}(\mathbf{s}_i, \mathbf{t}_k)/\tau)}{(\alpha \cdot p_i^T(y_i)) \sum_{\mathbf{t}_k \in P(i)} \exp(\text{sim}(\mathbf{s}_i, \mathbf{t}_k)/\tau) + \sum_{\mathbf{t}_j \in N(i)} (\beta \cdot (1 - p_i^T(y_j))) \cdot \exp(\text{sim}(\mathbf{s}_i, \mathbf{t}_j)/\tau)}\right)$$

3.2 What This Achieves: Rich Semantic Structuring with Full Decoupling

This formulation intelligently combines the strengths of SupCon and logit distillation. It uses the teacher's dark knowledge to build a semantically meaningful feature space while achieving **full decoupling** of target-class and non-target-class knowledge.

- **Adaptive Pull (Target-Class Knowledge):** By setting $w_{\text{pull}} = \alpha \cdot p_i^T(y_i)$, the pull force adapts to example difficulty:
  - **Easy examples:** Teacher confident ($p_i^T(y_i) \approx 1$) $\rightarrow$ Strong pull ($w_{\text{pull}} \approx \alpha$)
  - **Hard examples:** Teacher uncertain ($p_i^T(y_i) \approx 0.5$) $\rightarrow$ Moderate pull ($w_{\text{pull}} \approx 0.5\alpha$)
  - This prevents "over-pulling" on easy examples while still maintaining cluster cohesion
- **Intelligent, Non-Uniform Push (Non-Target-Class Knowledge):** The "push" force is weighted to create a semantic geometry:
  - Let the anchor $\mathbf{s}_i$ be a 'dog' ($y_i =$ 'dog').
  - **Confusing Negative (e.g., 'wolf'):** Let $\mathbf{t}_j$ be a 'wolf' vector ($y_j =$ 'wolf'). The teacher is confused, so its probability is high: $p_i^T(\text{'wolf'}) = 0.3$. The push weight is $w_{\text{push}} = \beta \cdot (1 - 0.3) = \beta \cdot 0.7$ (a **moderate** push).
  - **Irrelevant Negative (e.g., 'car'):** Let $\mathbf{t}_l$ be a 'car' vector ($y_l =$ 'car'). The teacher is confident this is not a car, so $p_i^T(\text{'car'}) = 0.001$. The push weight is $w_{\text{push}} = \beta \cdot (1 - 0.001) = \beta \cdot 0.999$ (a **very strong** push).
- **Rich Semantic Structuring:** The student's latent space is forced to learn the teacher's semantic similarity map. The loss explicitly penalizes closeness to 'car' (a distant class) **more** than it penalizes closeness to 'wolf' (a similar class). This directly achieves the goal of making $\text{sim}(\mathbf{s}_{\text{dog}}, \mathbf{t}_{\text{wolf}}) > \text{sim}(\mathbf{s}_{\text{dog}}, \mathbf{t}_{\text{car}})$.
- **Full Decoupling:** α controls target-class structure (intra-class compactness) while β controls non-target-class structure (inter-class semantic distances). These can now be tuned independently, with both utilizing the teacher's probability distribution to weight their respective contributions.
- **Tunable Control:** α and β act as global knobs to balance the overall importance of the "pull" (cluster tightness) vs. the "push" (semantic separation).

3.3 Solving the "Hard Positive" Problem with Adaptive Weighting

A known issue in standard SupCon is that the gradients are "coupled": the total pull force on positives mathematically equals the total push force on all negatives. This creates a problem with **hard positives**:

1. A single hard positive (where $\mathbf{s}_i$ is far from its positive $\mathbf{t}_k$) creates a *massive* pull gradient.
2. Due to the coupling, this also creates a *massive* total push gradient.
3. In standard SupCon, this huge push gradient is "wasted" by being distributed democratically and thinly across *all* negatives (e.g., 1000 of them). The push on any *individual* negative becomes uselessly small.

The logit-weighted approach with adaptive pull solves this problem more effectively.

- **Adaptive Pull on Hard Positives:** Hard positives often correspond to examples where the teacher is less confident. With $w_{\text{pull}} = \alpha \cdot p_i^T(y_i)$, the pull weight naturally adapts:

- If the teacher is uncertain ($p_i^T(y_i) = 0.6$), the pull is moderated ($w_{\text{pull}} = 0.6\alpha$)
- This prevents excessively large pull gradients while still maintaining directional guidance

- **Intelligent Push Redistribution:** The resulting push gradient is **diverted** based on semantic relevance:
  - Semantically **close** negatives get **small** weights (e.g., $p_i^T(\text{'wolf'}) = 0.3 \rightarrow w_{\text{push}} = 0.7\beta$)
  - Semantically **distant** negatives get **large** weights (e.g., $p_i^T(\text{'car'}) = 0.001 \rightarrow w_{\text{push}} = 0.999\beta$)

- **Efficient Gradient Utilization:** The available push gradient is concentrated where it matters most—pushing away truly irrelevant classes—while applying a gentler, more "respectful" push to semantically similar classes. This allows the feature space to develop proper semantic structure rather than forcing all non-target classes to be equally distant.

This adaptive weighting scheme achieves a highly logical outcome: gradients are allocated proportionally to semantic relevance, creating a feature space that naturally encodes the teacher's knowledge about both target-class cohesion and inter-class semantic relationships.