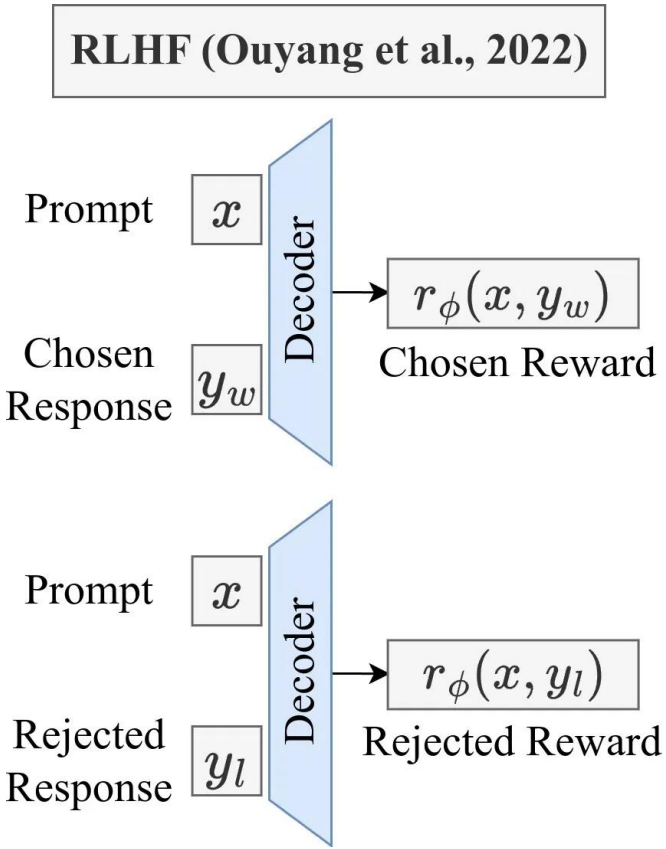
Swap-guided Preference Learning for Personalized Reinforcement Learning from Human Feedback

Gihoon Kim [1] **Euntai Kim** [1, 2]

[1] Yonsei University [2] Korea Institute of Science and Technology

gihoon@yonsei.ac.kr

Problem 1: RLHF overlooks individual preferences in reward modeling.

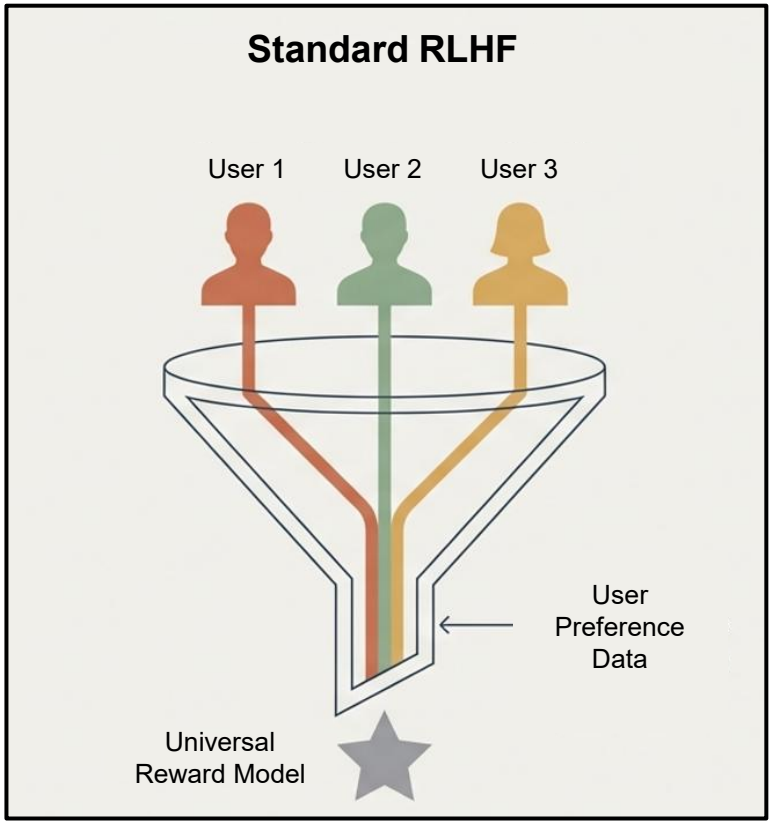


RLHF Objective:

Maximize $\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))$

A reward model is trained to predict which response people prefer.

Problem 1: RLHF overlooks individual preferences in reward modeling.

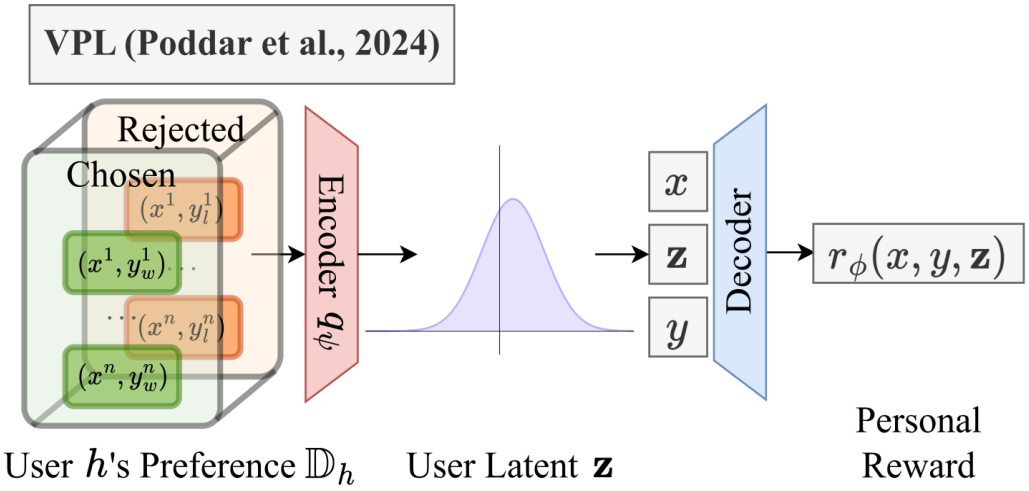


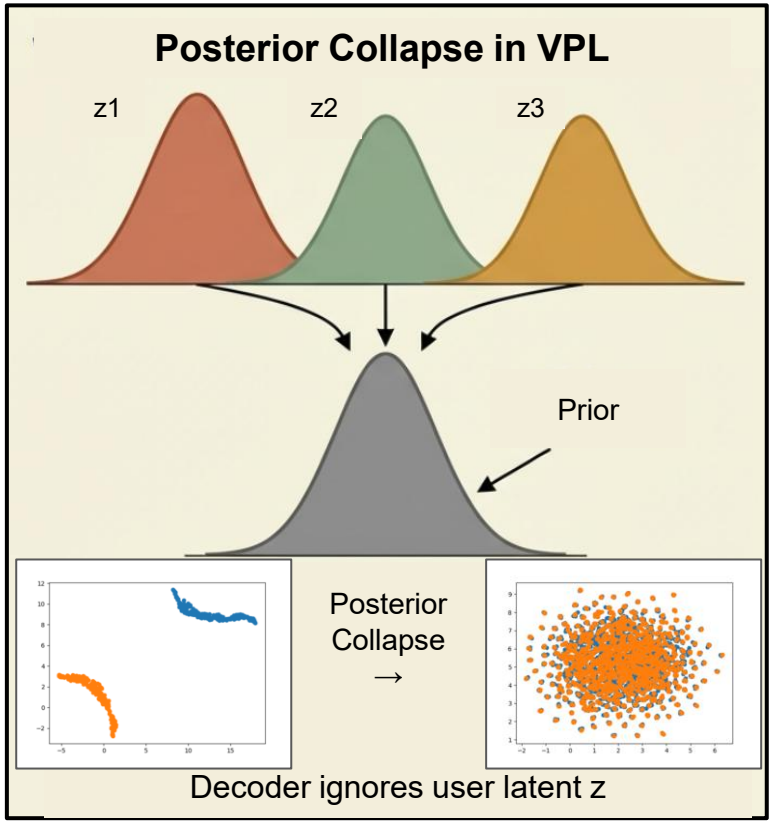
In reality, users value different things…

However, RLHF has one reward for everyone

→ Personalized alignment.

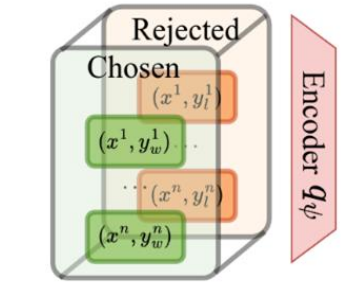
Problem 2: Posterior Collapse in Variational Preference Learning



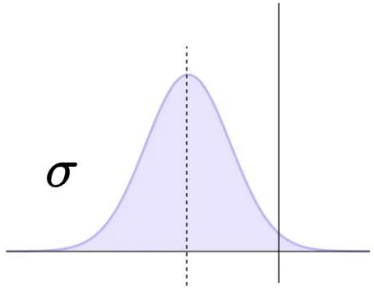


How can we prevent posterior collapse in preference learning?

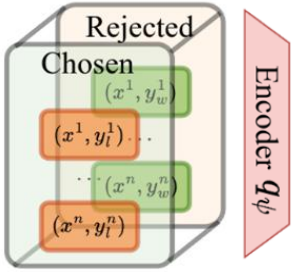
If user preferences are encoded correctly,
what should the opposite user look like in latent space?



User h's Preference $\mathbb{D}_h$



$$q_\psi(\boldsymbol{z} \mid \mathbb{D}_h) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$$

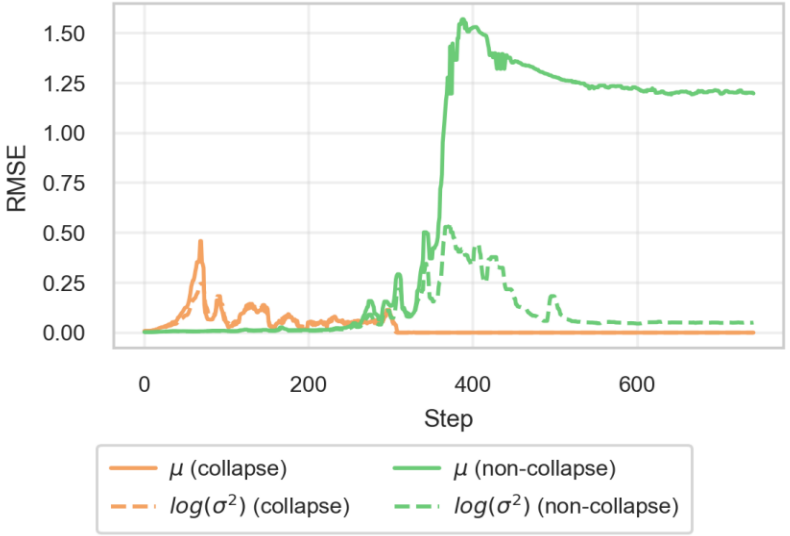


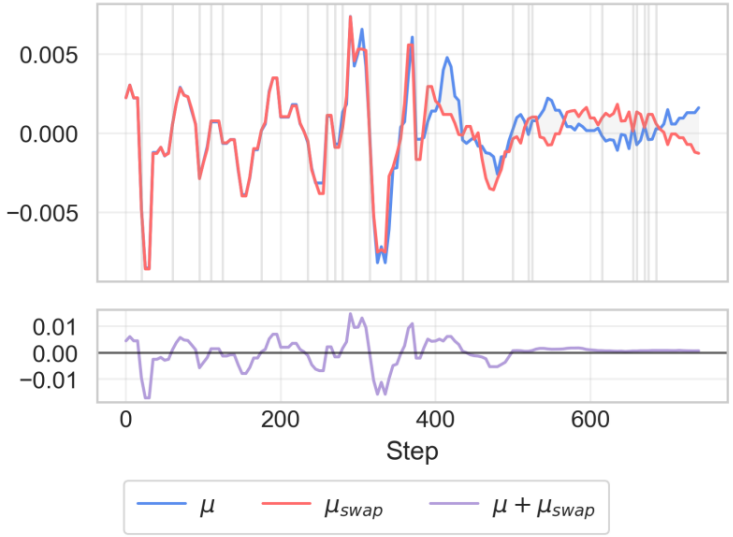
Swapped
User h's Preference $\mathbb{D}_{h_{\text{swap}}}$

?

$$q_\psi(\boldsymbol{z} \mid \mathbb{D}_{h_{\text{swap}}}) = \mathcal{N}(\boldsymbol{\mu}_{\text{swap}}, \boldsymbol{\sigma}^2_{\text{swap}})$$

How can we prevent posterior collapse in preference learning?

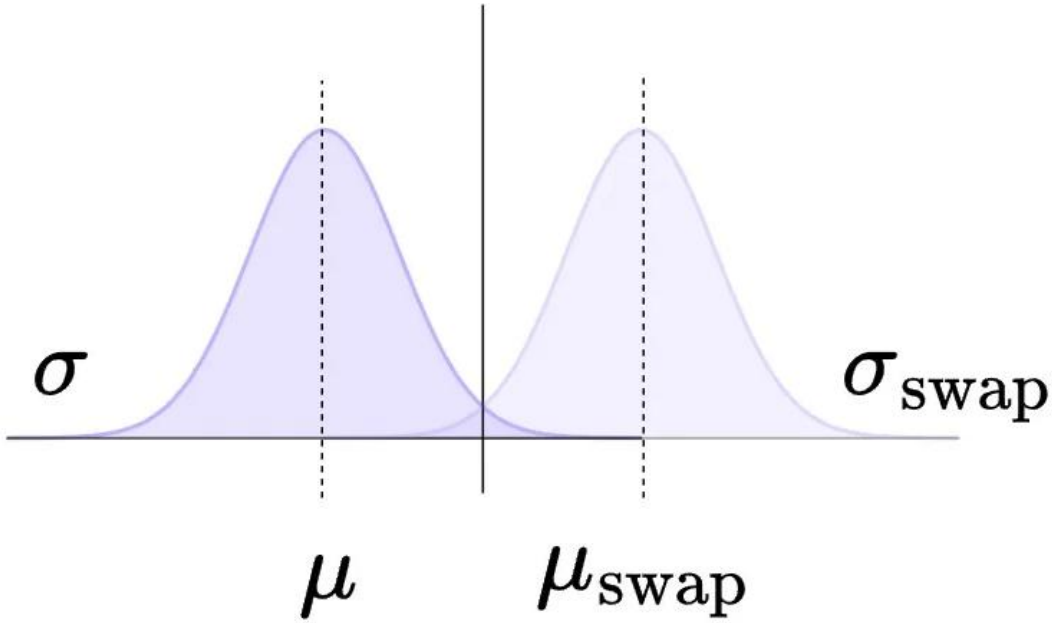




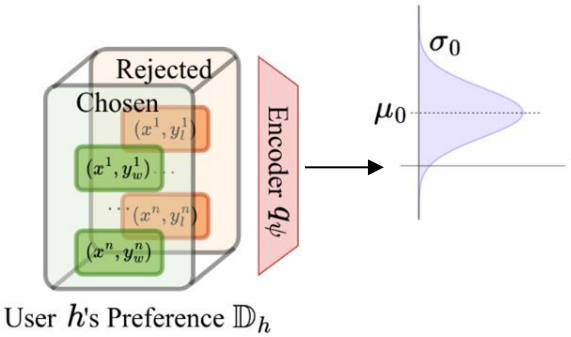
Collapse: $\mu \approx \mu_{\text{swap}}$ $\ell \approx \ell_{\text{swap}}$ $\star\, \ell = \log \sigma^2$

Non-collapse: $\mu \approx -\mu_{\text{swap}}$ $\ell \approx \ell_{\text{swap}}$

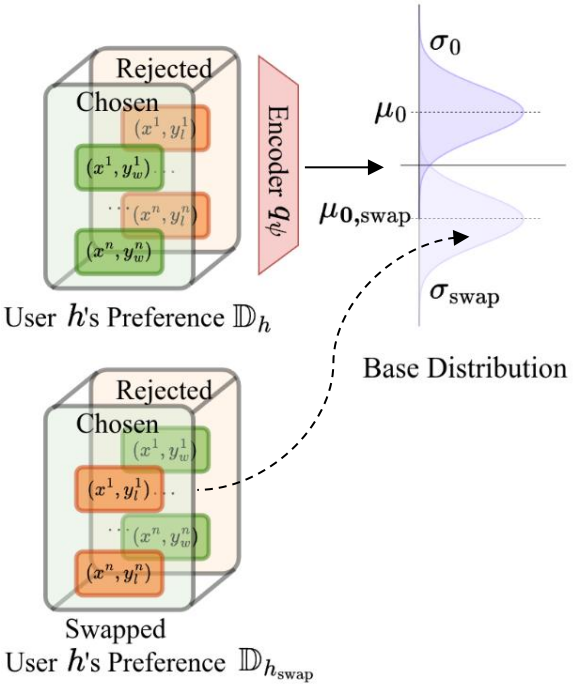
How can we prevent posterior collapse in preference learning?



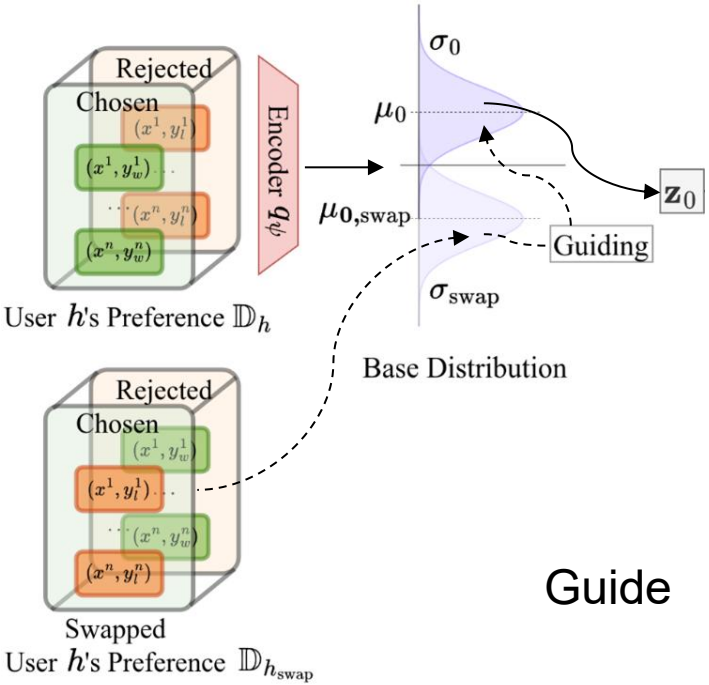
Key Idea 1: Swap-guided base regularization



Key Idea 1: Swap-guided base regularization

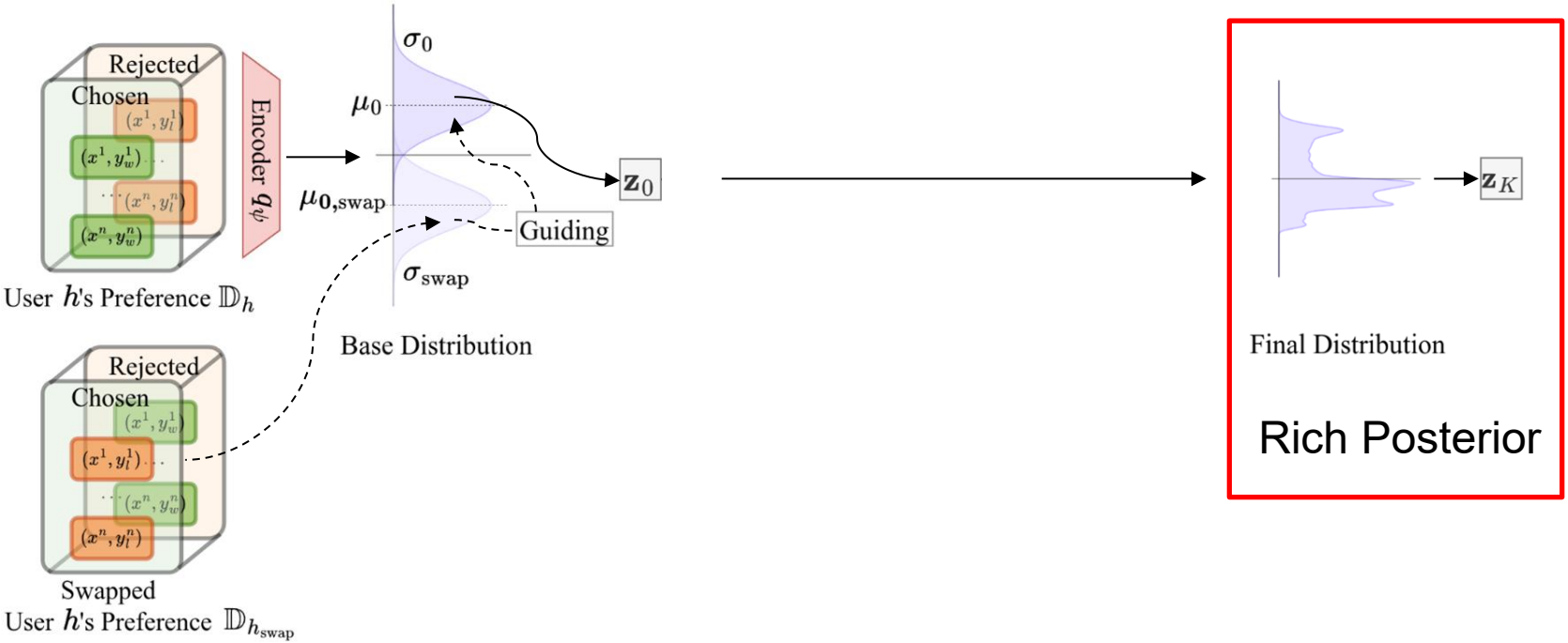


Key Idea 1: Swap-guided base regularization

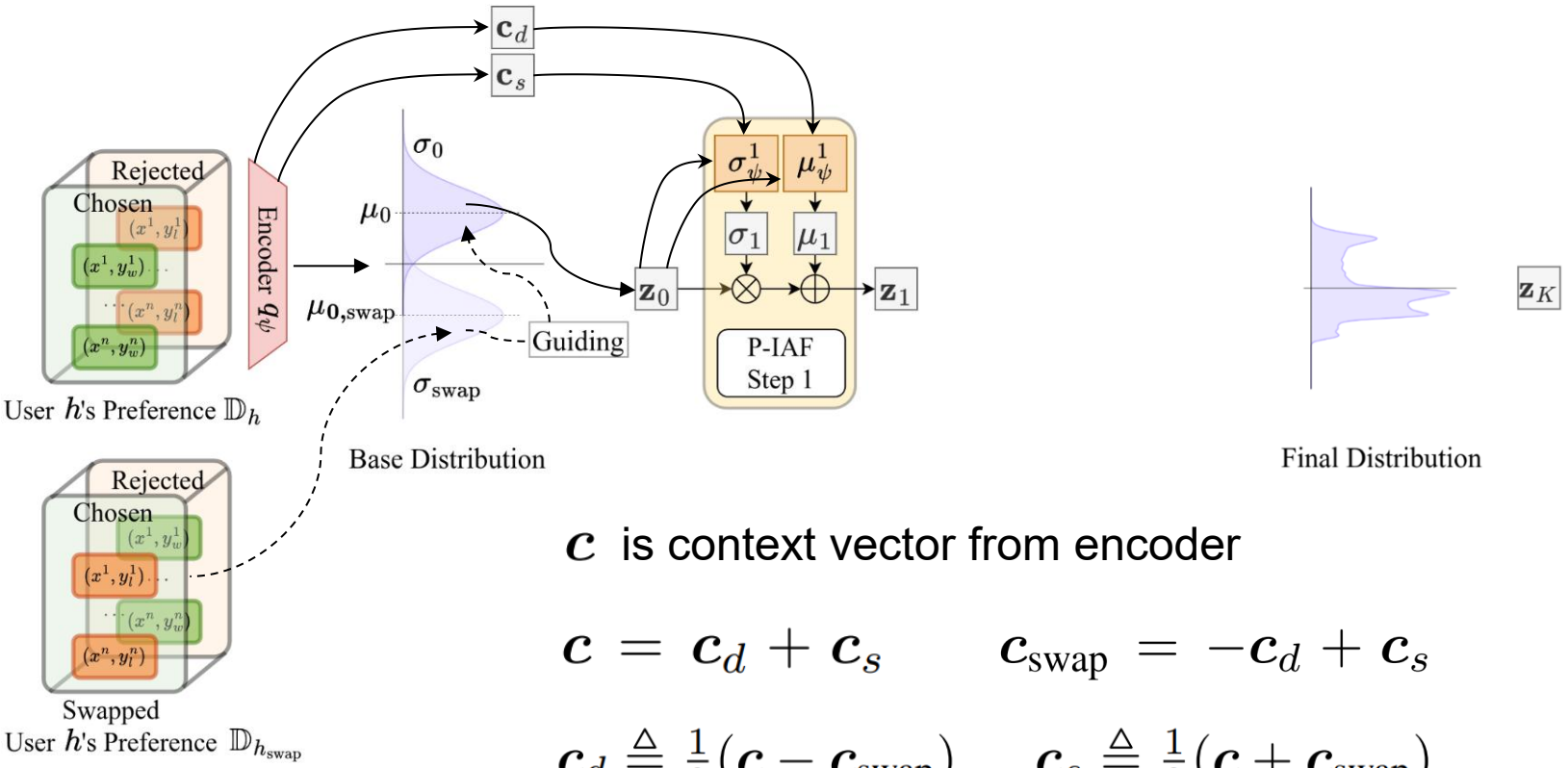


Guide $\quad \mu \approx -\mu_{\text{swap}} \quad \ell \approx \ell_{\text{swap}}$

Key Idea 2: Preferential-Inverse Autoregressive Flow



Key Idea 2: Preferential-Inverse Autoregressive Flow

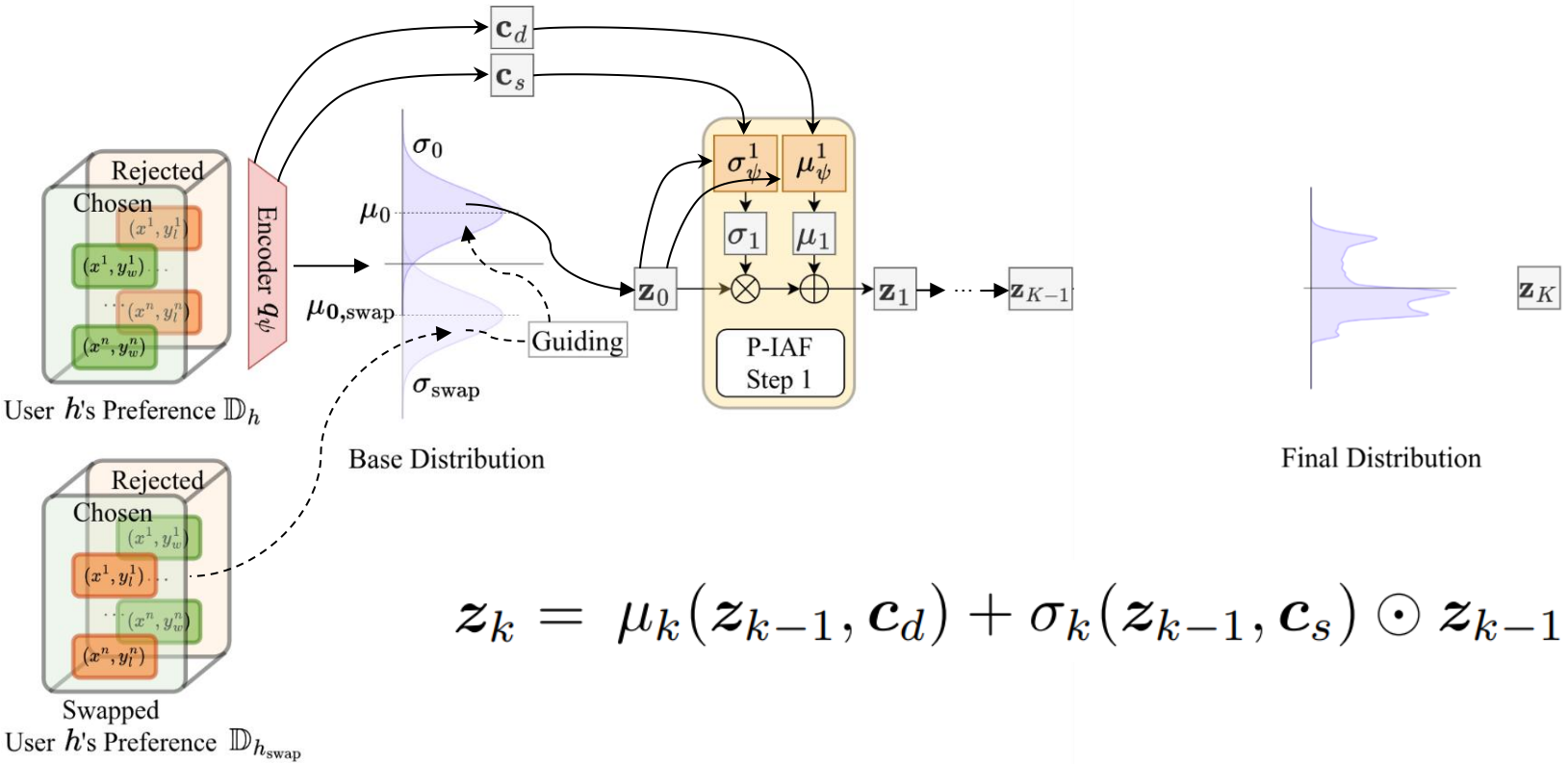


$\boldsymbol{c}$ is context vector from encoder

$$\boldsymbol{c} = \boldsymbol{c}_d + \boldsymbol{c}_s \qquad \boldsymbol{c}_{\text{swap}} = -\boldsymbol{c}_d + \boldsymbol{c}_s$$

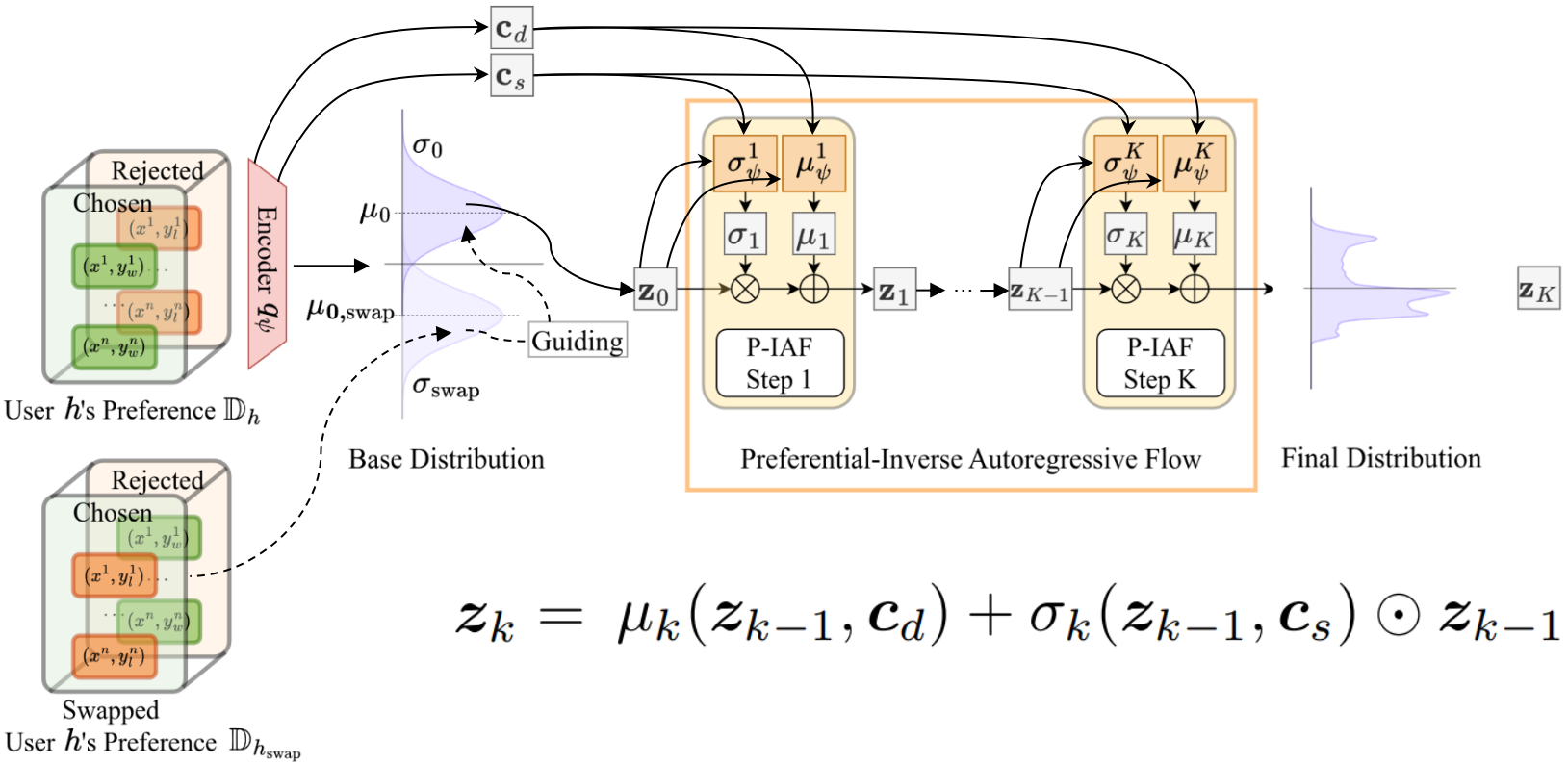
$$\boldsymbol{c}_d \triangleq \frac{1}{2}(\boldsymbol{c} - \boldsymbol{c}_{\text{swap}}), \quad \boldsymbol{c}_s \triangleq \frac{1}{2}(\boldsymbol{c} + \boldsymbol{c}_{\text{swap}})$$

Key Idea 2: Preferential-Inverse Autoregressive Flow



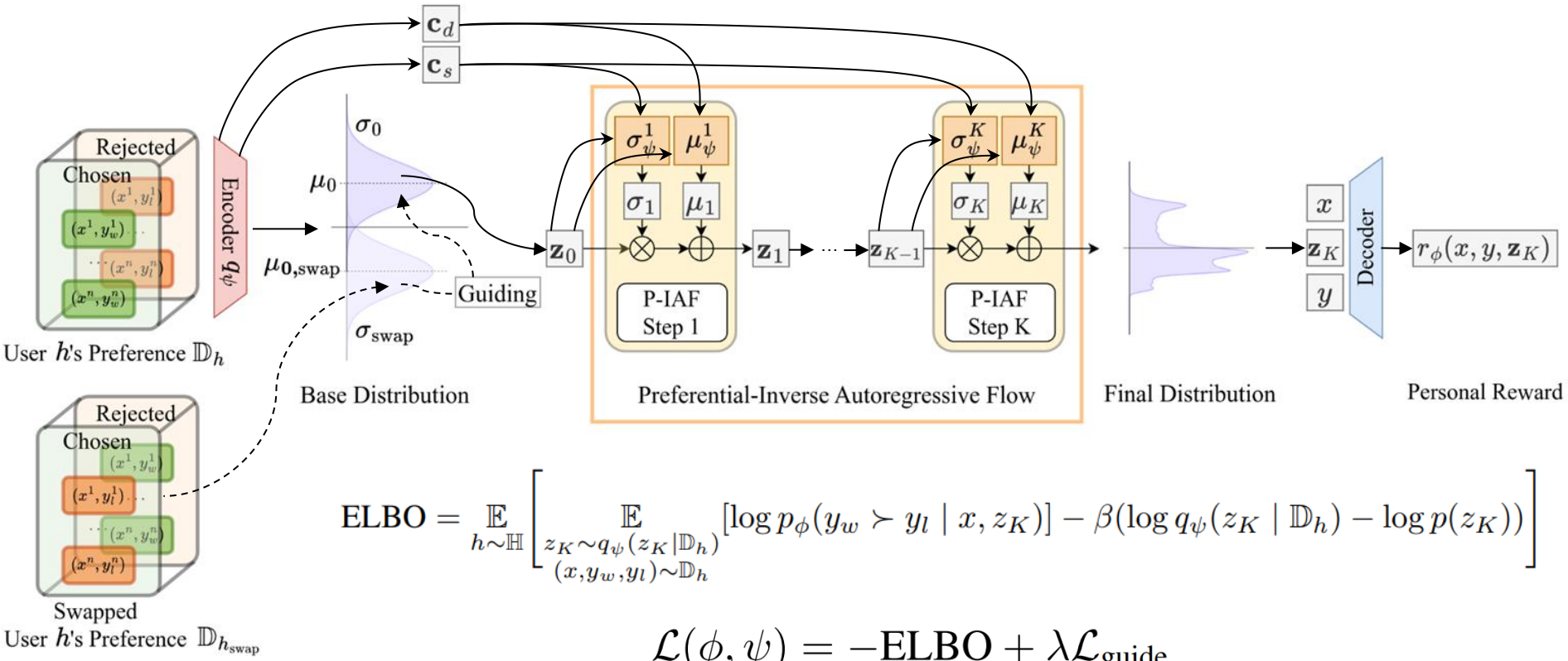
$$\boldsymbol{z}_k = \mu_k(\boldsymbol{z}_{k-1}, \boldsymbol{c}_d) + \sigma_k(\boldsymbol{z}_{k-1}, \boldsymbol{c}_s) \odot \boldsymbol{z}_{k-1}$$

Key Idea 2: Preferential-Inverse Autoregressive Flow



$$\boldsymbol{z}_k = \mu_k(\boldsymbol{z}_{k-1}, \boldsymbol{c}_d) + \sigma_k(\boldsymbol{z}_{k-1}, \boldsymbol{c}_s) \odot \boldsymbol{z}_{k-1}$$

Key Idea 3: Adaptive Latent Conditioning



$$\text{ELBO} = \mathop{\mathbb{E}}_{h\sim\mathbb{H}}\left[\mathop{\mathbb{E}}_{\substack{z_K\sim q_\psi(z_K|\mathbb{D}_h)\\(x,y_w,y_l)\sim\mathbb{D}_h}}[\log p_\phi(y_w \succ y_l \mid x, z_K)] - \beta(\log q_\psi(z_K \mid \mathbb{D}_h) - \log p(z_K))\right]$$

$$\mathcal{L}(\phi,\psi) = -\text{ELBO} + \lambda\mathcal{L}_{\text{guide}}$$

Results

VPL collapses for every tested β value.

In contrast, SPL maintains high AU across all settings.

Model	β	Method	UF-P-4	
			Acc. [%]	AU [%]
Llama-3.1-8B	3×10^{-7}	VPL	57.25 ± 0.22	0.00 ± 0.00
		SPL	61.92 ± 0.08	93.75 ± 4.67
	3×10^{-6}	VPL	57.14 ± 0.05	0.00 ± 0.00
		SPL	62.21 ± 0.06	$\mathbf{96.19 \pm 2.33}$
	3×10^{-5}	VPL	57.18 ± 0.11	0.00 ± 0.00
		SPL	$\mathbf{62.46 \pm 0.07}$	85.90 ± 3.40

*β: Influence of KL divergence

Results

SPL achieves the best accuracy among the compared baselines.

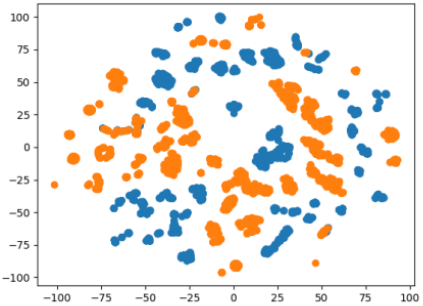
Table 2: Preference-prediction accuracy (%) compared with baselines

Model	Method	Pets	UF-P-2	UF-P-4
Llama-3.2-3B	BTL	57.48 ± 2.37	62.25 ± 0.03	57.07 ± 0.01
	DPL	62.02 ± 1.92	62.22 ± 0.03	57.04 ± 0.05
	VPL	99.67 ± 0.38	62.37 ± 0.15	57.03 ± 0.10
	SPL (Ours)	$\mathbf{100.0 \pm 0.00}$	$\mathbf{63.28 \pm 0.13}$	$\mathbf{61.56 \pm 0.03}$
Llama-3.1-8B	BTL	60.74 ± 0.49	62.59 ± 0.04	57.40 ± 0.28
	DPL	61.03 ± 0.25	62.74 ± 0.03	57.66 ± 0.14
	VPL	75.33 ± 0.63	62.66 ± 0.23	57.14 ± 0.05
	SPL (Ours)	$\mathbf{100.0 \pm 0.00}$	$\mathbf{63.71 \pm 0.18}$	$\mathbf{62.21 \pm 0.06}$

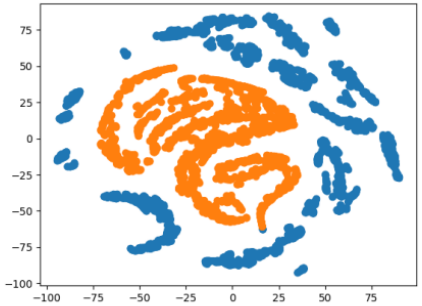
Results

Table 3: Training computation and memory costs on UF-P-4

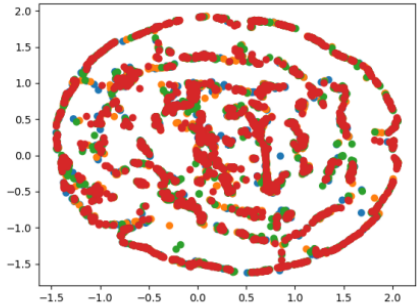
Model	Method	sample/s	GPU hour	peak memory
Llama-3.2-3B	VPL	6.070	13.363	6.25GB
	SPL	5.952	13.590	6.65GB
Llama-3.1-8B	VPL	3.370	23.791	11.37GB
	SPL	3.355	23.945	11.76GB



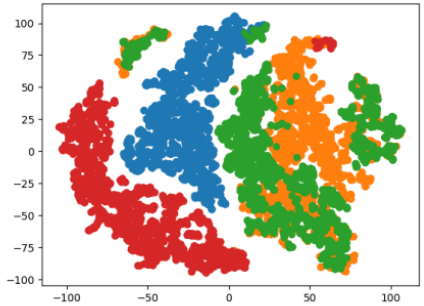
(a) VPL (UF-P-2)



(b) SPL (UF-P-2)



(c) VPL (UF-P-4)



(d) SPL (UF-P-4)

Conclusion

- SPL uses the structure induced by swapping preference pairs as a direct learning signal.
- This prevents posterior collapse and produces more identifiable and stable user latents.
- As a result, it enables more faithful personalized reward modeling.