# Enhancing Emotional Support Conversation with Plug-in Modules for Sequential Strategy Modeling and Strategy-Specialized Generation

**Yerin Park**[1,†] **Junhyeok Park**[2] **Junsu Cho**[2*]
{diluny2, eppi9999, jscho}@pusan.ac.kr
[1]Department of Information Convergence Engineering, Pusan National University
[2]School of Computer Science and Engineering, Pusan National University
[†]Current affiliation: Hanwha Systems

## Abstract

Emotional Support Conversation (ESC) aims to alleviate users' emotional distress through multi-turn interactions by selecting appropriate support strategies and generating aligned responses. Despite recent progress, existing approaches face three key limitations: (1) the inability to capture sequential transition patterns of strategies within conversations; (2) a training-inference mismatch in strategy prediction; and (3) reliance on a shared decoder that limits strategy-specific expression. To address these challenges, we propose S3-ESC, a simple framework of three components that are model-agnostic across encoder-decoder ESC backbones. First, we introduce a **S**equential strategy predictor that models strategy transition patterns. Second, we apply **S**cheduled sampling for strategy to mitigate the training-inference mismatch. Third, we propose **S**elective LoRA for strategy that enables strategy-specialized generation. Experiments on six ESC backbones show that S3-ESC consistently improves strategy prediction accuracy, by up to 4.5pp, while enhancing response quality on most of them. Our code is publicly available.[1]

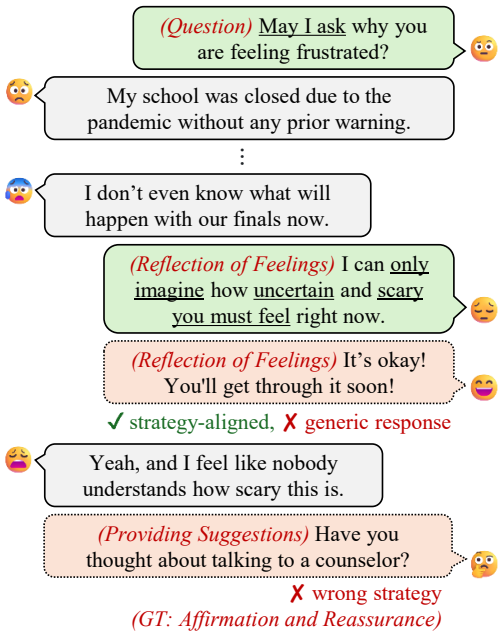



Figure 1: An example of an emotional support conversation between the support-seeker (left) and the supporter (right). Support strategies are shown in red italics, with strategy-relevant keywords underlined. The first red dashed box shows a generic response under the correct strategy, while the second shows an incorrect strategy prediction leading to premature advice.

## 1 Introduction

Emotional Support Conversation (ESC) is a goal-directed dialog task that aims to reduce users' emotional distress through multi-turn interactions (Liu et al., 2021). Unlike empathetic chatting, which centers on expressing understanding, ESC requires selecting an appropriate *support strategy* at each turn and generating a response that realizes it. Grounded in Hill's Helping Skills Theory (Hill, 1999), the ESC Framework (Liu et al., 2021) defines support strategies such as *Question*, *Self-disclosure*, and *Providing Suggestions*, organized into stages that guide the conversation from exploration to action. An effective ESC system therefore couples two capabilities: predicting an appropriate strategy for each turn, and generating a response that aligns with it. Since the predicted strategy conditions the subsequent response, strategy prediction plays a central role in the ESC pipeline.

**Challenges.** Various methods have improved strategy prediction and response generation, but several key limitations remain. First, strategy prediction does not sufficiently capture the patterns of strategy transitions within a conversation. In our experiments under a unified evaluation setting, most existing models achieve strategy prediction

* Corresponding author.
[1]Our code is available at https://github.com/Yerin99/S3-ESC.



*Findings of the Association for Computational Linguistics: EMNLP 2026*, pages 37365–37379
October 24-29, 2026 ©2026 Association for Computational Linguistics

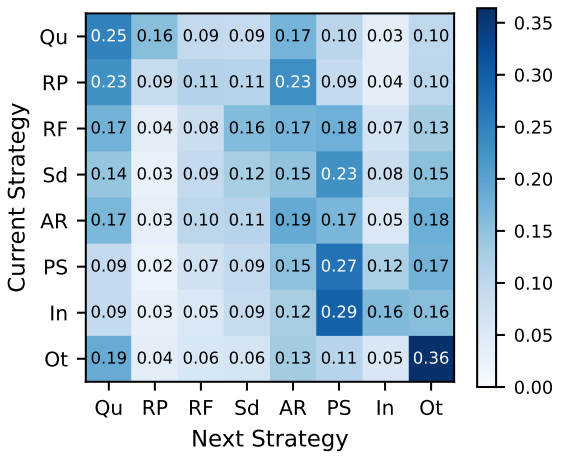



Figure 2: Strategy transition probabilities on ESConv. Strategy abbreviations are defined in Appendix A.

accuracies of only 30–34% over the eight strategies, including 30.85% for MISC (Tu et al., 2022), 33.67% for DKFPE (Hao and Kong, 2025), and even lower performance for large language models (e.g., GPT-4 at 22.1%; Kang et al., 2024). Our analysis of strategy transitions on the ESConv dataset reveals that strategy sequences exhibit regular patterns, such as *Question* → *Affirmation and Reassurance* → *Question*. As shown in Fig. 2, certain strategies frequently follow specific predecessors, suggesting that past strategy sequences contain predictive information for the next strategy. Table 3 supports this observation: a sequential model trained to predict the next strategy using only previous strategy sequences achieves accuracy comparable to existing ESC models that leverage full dialog context. Nevertheless, existing methods either overlook such transition patterns, or exploit them only in a limited manner, as prior approaches often discard ordering information through coarse aggregation (Peng et al., 2023) or restrict transition modeling to local dependencies within a fixed-size window (Zhao et al., 2023).

Second, the predicted strategy directly conditions response generation, so prediction errors carry through a training-inference mismatch: during training, as in teacher forcing, the decoder is conditioned on the ground-truth strategy, whereas at inference it must rely on its own prediction. Since the decoder is trained only under correct strategy conditions, incorrect strategy prediction can steer generation toward inappropriate response directions.

Third, existing models remain limited in how they realize the predicted strategy in the generated response. Current approaches incorporate strategies through input-level conditioning mechanisms, such as prepending strategy tokens (Liu et al., 2021), or passing strategy representations via cross-attention (Tu et al., 2022). Despite their differences, these approaches all fall under *strategy-conditioned* generation, where the strategy influences response generation as a condition while the decoder parameters remain shared across all strategies. However, prior work shows that each strategy is significantly associated with distinct vocabulary patterns ($z$-score $> 3$) (Liu et al., 2021). For example, *Question* with interrogative phrases (e.g., do you, how), *Providing Suggestions* with hedging and advisory expressions (e.g., maybe, have you), and *Reflection of Feelings* with empathetic expressions (e.g., understand how, can tell). This suggests that a shared decoder may struggle to capture the distinct mode of expression of each strategy through conditioning alone, and motivates a more *strategy-specialized* approach where the decoder parameters are partially specialized per strategy.

**Approach.** To this end, we propose S3-ESC, a plug-in framework for improved strategy prediction and strategy-specialized response generation, model-agnostic across encoder-decoder ESC backbones, built from three simple components. First, to exploit transition patterns, we introduce a ***S**equential strategy predictor* that models past strategies with a self-attentive architecture (Kang and McAuley, 2018) and ensembles its prediction with the base predictor. Second, to reduce the training-inference mismatch, we apply ***S**cheduled sampling for strategy* (Bengio et al., 2015), which exposes the decoder to predicted strategies during training. Third, for strategy-specialized generation, we propose ***S**elective LoRA for strategy*, which augments a shared decoder with per-strategy LoRA experts (Hu et al., 2022), letting each expert capture strategy-specific patterns. These three components are complementary: the sequential predictor improves strategy prediction, scheduled sampling reduces the training-inference mismatch, and the resulting strategy signal routes the appropriate LoRA expert for strategy-specialized generation. Experiments across six backbones show consistent gains in strategy prediction while improving response quality for most backbones. The ablation study (§4.3), the strategy sequence analysis (§4.4), the turn-wise analysis (§4.5), and the human evaluation (§4.6) examine the contribution of each component.

Our contributions are as follows:

- We identify three limitations of existing ESC systems: unexploited strategy transitions, a training-inference mismatch, and a shared decoder that limits strategy-specific expression.
- We propose S3-ESC, a plug-in framework, model-agnostic across encoder-decoder ESC backbones, that addresses these limitations with a sequential strategy predictor, scheduled sampling for strategy tokens, and selective LoRA.
- We evaluate S3-ESC on six ESC backbones. Strategy prediction accuracy improves on all six, and response quality improves on most of them.

## 2 Related Work

### 2.1 Emotional Support Conversation

Emotional Support Conversation (ESC) aims to reduce help-seekers' emotional distress through strategic multi-turn interactions (Liu et al., 2021), coupling strategy prediction with response generation at each turn. BlenderBot-Joint (Liu et al., 2021) introduced this strategy-conditioned formulation, prepending a predicted strategy token to the decoder input and generating the response conditioned on it. Subsequent approaches incorporate external knowledge into this formulation: MISC (Tu et al., 2022) integrates commonsense knowledge from COMET and represents the strategy as a mixture distribution over a codebook; KEMI (Deng et al., 2023) retrieves case knowledge from a mental health knowledge graph for mixed-initiative response generation; and DKFPE (Hao and Kong, 2025) dynamically filters commonsense knowledge and extracts persona information. PMPC (Xu et al., 2025) extracts multiperspective cues from dialog history and constructs semantic enhancement and constraint prompts. GLHG (Peng et al., 2022) employs a global-to-local hierarchical graph network to capture both global emotion causes and local user intentions, without a strategy predictor.

### 2.2 Strategy Utilization in Response Generation

Existing approaches integrate the predicted strategy into response generation through various mechanisms, such as prepending it as a decoder token (Liu et al., 2021), representing it as a codebook distribution (Tu et al., 2022), or injecting it into encoder representations (Peng et al., 2023). Although these mechanisms differ, they remain limited by relying on a single shared decoder to generate responses for diverse strategies with different utterance patterns. Recent work has adopted LoRA for efficient fine-tuning of LLMs in ESC (Zhao et al., 2025; Zheng et al., 2024; Zhang et al., 2025), but also without strategy-level differentiation. In contrast, we move beyond strategy-conditioned generation by adopting a strategy-specialized formulation, where the decoder parameters are partially differentiated across strategies.

## 3 Method

### 3.1 Task Definition

Given a dialog history $C_t = (u_1, u_2, \ldots, u_{2t-1})$ between a help-seeker and a supporter at turn $t$, the ESC task requires the system to (1) predict a support strategy $s_t$ from a predefined set $\mathcal{S} = \{s^1, \ldots, s^K\}$ of $K$ strategies, and (2) generate a response $R_t = (r_1, \ldots, r_L)$ conditioned on $C_t$ and $s_t$. Our method, following most existing ESC models, realizes this as a two-step pipeline: a strategy predictor estimates $\hat{s}_t$ from the dialog context, and a conditional generator produces $R_t$ given $C_t$ and $\hat{s}_t$.

### 3.2 Overview

Fig. 3 illustrates the overall architecture. S3-ESC augments an encoder-decoder backbone, such as BlenderBot (Liu et al., 2021), with three model-agnostic components: Selective LoRA (§3.3), Scheduled sampling for strategy (§3.5), and a SASRec-based strategy sequence ensemble (§3.6). All three rely on a strategy prediction signal $p_{str}$, obtained from either the backbone's existing classifier or a lightweight linear probe (§3.4). These components can be applied as plug-in modules on top of existing encoder-decoder ESC frameworks.

During training, S3-ESC lets the encoder process the dialog context and produce hidden states $\mathbf{h}_{\text{enc}}$. The strategy predictor produces a strategy distribution $\mathbf{p}_{\text{str}}$ from $\mathbf{h}_{\text{enc}}$ (§3.4). Then S3-ESC selects a strategy to feed to the decoder according to a predefined probability, choosing between the ground-truth strategy and the predictor's own prediction (§3.5). The selected strategy determines the appropriate LoRA expert in the decoder (§3.3), which is then used to generate the response.

During inference, the strategy predictor produces $\mathbf{p}_{\text{str}}$, while our sequential strategy predictor independently predicts a strategy distribution $\mathbf{p}_{\text{seq}}$ from the historical strategy sequence. The ensemble dis-

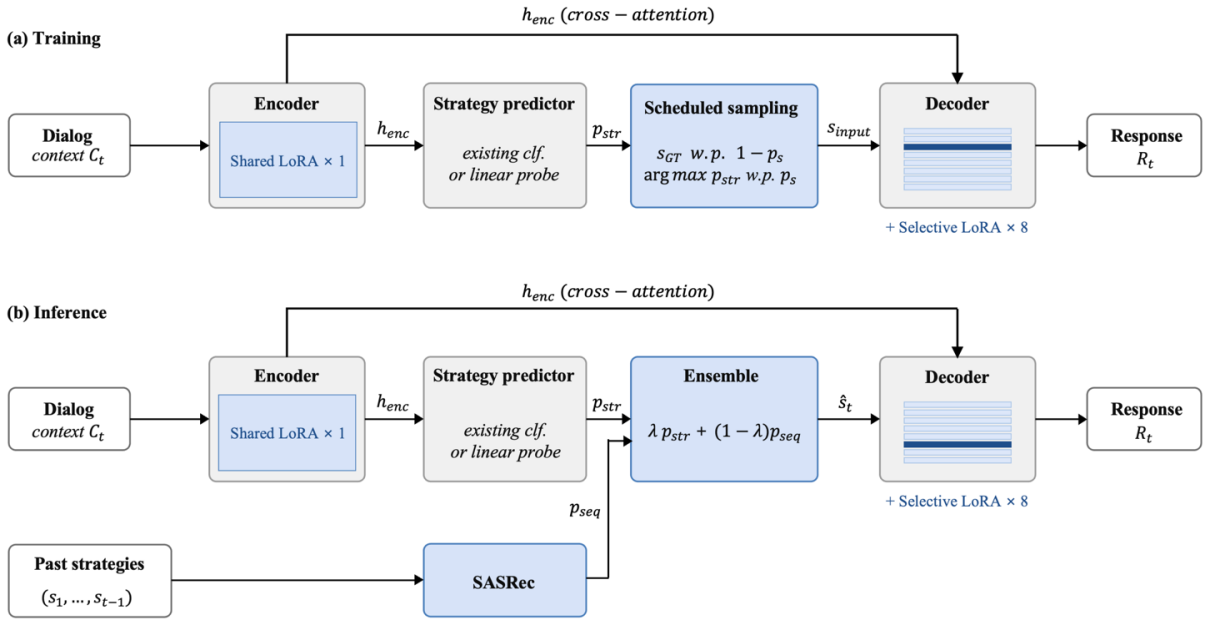



Figure 3: Overview of S3-ESC. (a) During training, scheduled sampling selects the strategy fed to the decoder, either the ground-truth strategy or the predictor's own prediction. (b) During inference, the sequential predictor (SASRec), pretrained independently and reused without retraining, is ensembled with the strategy predictor. In both stages the selected strategy routes the corresponding LoRA expert in the decoder.

tribution $\mathbf{p}_{\text{final}} = \lambda \cdot \mathbf{p}_{\text{str}} + (1-\lambda) \cdot \mathbf{p}_{\text{seq}}$ is used to select the final strategy, which in turn routes the corresponding LoRA expert for generation.

### 3.3 Selective LoRA for Strategy

As mentioned above, existing ESC models employ strategy-conditioned generation, where the strategy modulates the decoder's input or attention while the decoder parameters remain identical across strategies, limiting their ability to generate responses with strategy-specific keywords and tones. To address this limitation, we propose *selective LoRA for strategy*, which partially specializes the decoder parameters across strategies by augmenting the decoder with strategy-specific LoRA modules, thereby enabling strategy-specialized generation.

Following LoRA (Hu et al., 2022), we augment each weight matrix in decoder with a low-rank decomposition. For a given strategy $s$, the output of a linear layer becomes:

$$\mathbf{h}^{(l,p)} = W_0^{(l,p)}\mathbf{x} + \frac{\alpha}{r} B_s^{(l,p)} A_s^{(l,p)}\mathbf{x} \qquad (1)$$

where $l$ indexes the decoder layer and $p$ indexes one of the six projections in that layer: query, key, value, output, and the two projections in point-wise feed-forward layers. $W_0^{(l,p)} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ is the corresponding pretrained weight matrix, $A_s^{(l,p)} \in \mathbb{R}^{r \times d_{\text{in}}}$ and $B_s^{(l,p)} \in \mathbb{R}^{d_{\text{out}} \times r}$ are the low-rank matrices specific to strategy $s$, $r$ is the rank, and $\alpha$ is the scaling factor. Each strategy therefore maintains its own low-rank pair for every projection in every decoder layer. We refer to each such set of low-rank matrices as an *expert*, following the mixture-of-LoRA-experts literature (Dou et al., 2024; Wu et al., 2024), where the term denotes a LoRA module associated with a distinct domain or capability. In our setting the specialization is to a support strategy, and which expert is used is determined by the predicted strategy rather than by a learned gating function. We jointly fine-tune the LoRA experts and the backbone model, so that $W_0$ is updated together with the experts. Since $W_0$ is shared across all strategies while each expert is used only for its own strategy, $W_0$ can capture the conversational competence common to all strategies, while each LoRA expert captures strategy-specific generation patterns. For encoder, we use a single shared LoRA without strategy routing, adapting encoder representations without strategy-level specialization.

### 3.4 Strategy Predictor

Selective LoRA for strategy requires an explicit strategy prediction to select the appropriate expert. We denote the output of the strategy predictor as $\mathbf{p}_{\text{str}} \in \mathbb{R}^K$, a probability distribution over the $K$ strategies. In our framework, how $\mathbf{p}_{\text{str}}$ is obtained

depends on the backbone architecture.

Some ESC models already include an explicit strategy classifier that is jointly trained with the generation objective (Tu et al., 2022; Zhao et al., 2023; Xu et al., 2025). For these backbones, we retain the existing classifier and use its output directly as $\mathbf{p}_{\text{str}}$, requiring no architectural modification for strategy prediction.

For backbone models that lack a dedicated strategy classifier, to minimize interference with the backbone, we employ a linear probe on the encoder hidden states as the strategy predictor and stop gradients from propagating into the backbone. Although a decoder-side probe is also applicable, we find that the encoder-side probe yields higher accuracy, as shown in Appendix E. We obtain the input for linear probing, $\mathbf{h}_{\text{str}}$, using the attention-masked mean pool of the encoder hidden states:

$$\mathbf{h}_{\text{str}} = \frac{\sum_{i=1}^{L_{\text{in}}} \mathbf{h}_i^{\text{enc}} \cdot m_i}{\sum_{i=1}^{L_{\text{in}}} m_i} \tag{2}$$

where $\mathbf{h}_i^{\text{enc}} \in \mathbb{R}^d$ is the encoder output at position $i$, $m_i \in \{0, 1\}$ is the attention mask, and $L_{\text{in}}$ is the input sequence length. The strategy distribution is then:

$$\mathbf{p}_{\text{str}} = \text{softmax}(W_p\, \mathbf{h}_{\text{str}} + \mathbf{b}_p) \tag{3}$$

where $W_p \in \mathbb{R}^{K \times d}$ and $\mathbf{b}_p \in \mathbb{R}^K$.

**Isolated Training for Linear Probe.** The probe operates on detached hidden states: $\mathbf{h}_{\text{str}}$ is detached from the computation graph before being passed to the probe. This ensures that the backbone is trained solely by the generation objective, without being affected by the strategy classification loss. The strategy classifier via linear probing is trained with a separate optimizer from the backbone, allowing it to converge independently.

**Two-stage Training.** We adopt a two-stage training procedure for the strategy probe. In the first stage, the probe is trained alongside the backbone on gradient-blocked hidden states to provide online strategy predictions required for scheduled sampling, which stochastically replaces the ground-truth strategy with the model's own prediction during training. In the second stage, after backbone training is completed, we further train only the probe on hidden states extracted from the trained backbone to maximize strategy prediction accuracy. This two-stage procedure ensures both effective scheduled sampling during training and strong strategy prediction performance.

### 3.5 Scheduled Sampling for Strategy

During training, existing methods feed the ground-truth strategy to the decoder, whereas at inference time the model must rely on its own predicted strategy. This mismatch causes errors in strategy prediction to propagate to response generation, leading to degraded response quality as a result. To overcome this limitation and enable the model to generate appropriate responses even when the predicted strategy is incorrect, we apply scheduled sampling (Bengio et al., 2015) for LoRA selection.

When training, the strategy to feed to the decoder, $s_{\text{input}}$, is selected as:

$$s_{\text{input}} = \begin{cases} s_{\text{GT}} & \text{with probability } 1 - p_s \\ \arg\max(\mathbf{p}_{\text{str}}) & \text{with probability } p_s \end{cases} \tag{4}$$

where $s_{\text{GT}}$ is the ground-truth strategy, $p_s$ is the sampling probability and $\mathbf{p}_{\text{str}}$ is the predicted probability distribution for strategy from strategy predictor. We use a step schedule: $p_s = 0$ during the first epoch (i.e., pure teacher forcing), followed by a fixed value for subsequent epochs. This warmup phase allows the backbone and strategy predictor to stabilize before introducing imperfect strategies through sampling. By being exposed to incorrect strategies during training, the decoder learns to produce reasonable responses even when the predicted strategy does not match the ground truth, reducing the effect of the training-inference mismatch at inference time.

### 3.6 Strategy Prediction Ensemble with Sequential Modeling

Existing strategy predictors rely on the full dialog context, which makes it difficult to capture transition patterns in the strategy sequence, such as recurring cycles or dialog-phase transitions. We address this by ensembling the predictor with an external sequential model for strategy.

To this end, we adopt a self-attentive sequential model following the architecture of SASRec (Kang and McAuley, 2018), a self-attentive sequential recommendation model. Our sequential model formulates next-strategy prediction as a next-item prediction task over the chronologically ordered sequence of past strategies $(s_1, \ldots, s_{t-1})$. Following SASRec, it consists of a token embedding layer, learnable position embeddings, and a Transformer-based architecture with causal masking, and outputs a probability distribution $\mathbf{p}_{\text{seq}}$ over the $K$ strategies.

**Model-Agnostic Property.** Our sequential model uses only the strategy sequence as input. It does not access any backbone hidden states, encoder outputs, or dialog text. This makes it fully model-agnostic: a single trained model produces identical predictions regardless of which backbone is used, and can be shared across all backbones without retraining.

**Ensemble Mechanism.** During inference, the final probability distribution for predicting the strategy is a weighted average of $\mathbf{p}_{\text{str}}$ and $\mathbf{p}_{\text{seq}}$:

$$\mathbf{p}_{\text{final}} = \lambda \cdot \mathbf{p}_{\text{str}} + (1 - \lambda) \cdot \mathbf{p}_{\text{seq}} \tag{5}$$

where $\lambda \in [0, 1]$ is a mixing coefficient selected via grid search on the validation set, optimizing for strategy prediction accuracy. The predicted strategy is $\hat{s}_t = \arg\max(\mathbf{p}_{\text{final}})$.

The backbone strategy predictor and our sequential predictor provide complementary signals: the backbone predictor captures turn-level semantic cues from the dialog context, while the sequential predictor captures transition patterns across the conversation. By combining these signals, the ensemble can account for both contextual semantics and sequential dynamics that are not fully captured by either component alone, enabling more accurate strategy prediction.

### 3.7 Model Training

Our method does not modify the backbone model's training objective. The backbone is trained with its original generation loss $\mathcal{L}_{\text{gen}}$, which typically combines response generation (e.g., negative log-likelihood) with baseline-specific auxiliary losses such as strategy prediction or emotion classification. For backbones where we add a linear probe (§3.4), the probe is trained with a cross-entropy loss over the strategy labels:

$$\mathcal{L}_{\text{str}} = -\log \mathbf{p}_{\text{str}}(s_{\text{GT}} \mid \mathbf{h}_{\text{str}}). \tag{6}$$

For backbones that already include a strategy classifier, strategy prediction is handled by the baseline's own training objective within $\mathcal{L}_{\text{gen}}$.

Notably, most of our modules do not affect the training of backbone model parameters. Training the linear probe is detached from the computation graph; the two losses are thus computed in parallel and optimized by separate optimizers rather than jointly. Also the sequential strategy predictor is trained independently on strategy sequences prior to backbone training, and its parameters are frozen at inference time. As a result, the training dynamics of the backbone remain identical to the original model, except for scheduled sampling and selective LoRA. This decoupled design ensures the model-agnostic property of S3-ESC across encoder-decoder ESC backbones.

## 4 Experiments

### 4.1 Experimental Setup

**Dataset.** We conduct experiments on the ESConv dataset (Liu et al., 2021), which contains 1,300 multi-turn emotional support conversations annotated with eight support strategies, so that $K = 8$ in our experiments. We split the dataset at the conversation level with a 70:15:15 ratio for training, validation, and test, respectively.

**Strategy history.** Following the standard ESConv protocol (Liu et al., 2021), the strategy labels of preceding supporter turns are part of the observed dialog history: BlenderBot-Joint prepends the gold strategy token to every supporter utterance in the flattened input, and the other backbones condition on the same labels through their own mechanisms. Only the current-turn strategy is predicted. Our sequential predictor takes this same gold strategy history as input, so all models in Table 1 are evaluated under identical conditions.

**Baselines.** We evaluate six ESC models spanning a range of architectures: BlenderBot-Joint (Liu et al., 2021), GLHG (Peng et al., 2022), MISC (Tu et al., 2022), KEMI (Deng et al., 2023), PMPC (Xu et al., 2025), and DKFPE (Hao and Kong, 2025). We reproduce each baseline in our unified evaluation setting before applying S3-ESC.

**Metrics.** We evaluate each model in three aspects. For strategy prediction, we report accuracy (ACC). For generation quality, we report perplexity (PPL) and BLEU-$n$ ($n \in \{1, 2, 3, 4\}$) (Papineni et al., 2002). For generation diversity, we report Dist-$n$ ($n \in \{1, 2, 3\}$) (Li et al., 2016). We also report ROUGE-L (Lin, 2004) and token-level F1.

**Implementation details.** All experiments are conducted on a single NVIDIA A5000 GPU (24GB). For each baseline, we retain its original training and inference hyperparameters and only add the plug-in components of S3-ESC. The scheduled sampling probability $p_s$ is selected per backbone from $\{0.05, 0.10, 0.15, 0.20\}$; we analyze its

Table 1: Main results on the ESConv test set. All baselines are reproduced under a unified evaluation setting (conversation-level 70:15:15 split, checkpoint selected by lowest validation perplexity). In each baseline pair, the better result is **bolded**. GLHG does not include a strategy predictor; ACC is therefore not reported.

| Model | ACC↑ | PPL↓ | D-1↑ | D-2↑ | D-3↑ | B-1↑ | B-2↑ | B-3↑ | B-4↑ | R-L↑ | F1↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| BlenderBot (Joint) | 31.27 | 16.04 | **3.47** | 20.06 | 41.98 | 19.10 | 7.41 | 3.63 | 2.05 | 16.57 | 19.85 |
| +S3-ESC | **35.77** | **16.02** | 3.36 | **20.18** | **42.20** | **19.31** | **7.71** | **3.88** | **2.27** | **16.91** | **20.20** |
| GLHG | – | 16.31 | **3.88** | 22.32 | 46.20 | 17.26 | 6.49 | **3.24** | **1.84** | 16.18 | 19.13 |
| +S3-ESC | 35.17 | **15.96** | 3.72 | **22.78** | **47.46** | **17.34** | **6.51** | 3.16 | 1.77 | **16.34** | **19.29** |
| MISC | 30.85 | 17.39 | 2.59 | 12.04 | 21.90 | **18.41** | **7.79** | 4.18 | 2.49 | 17.45 | **20.42** |
| +S3-ESC | **34.24** | **17.08** | **2.63** | **12.37** | **22.99** | 17.69 | 7.66 | **4.22** | **2.59** | **17.63** | 20.41 |
| KEMI | 32.85 | 15.52 | 2.52 | 12.64 | 24.72 | 20.83 | 8.52 | 4.46 | 2.66 | 17.66 | 20.70 |
| +S3-ESC | **35.31** | **15.49** | **2.72** | **13.46** | **26.08** | **21.01** | **8.73** | **4.59** | **2.75** | **17.73** | **20.80** |
| PMPC | 32.10 | 15.70 | **3.92** | **23.15** | **47.38** | 18.00 | 7.03 | 3.55 | 2.03 | 16.73 | 19.93 |
| +S3-ESC | **35.13** | **15.66** | 3.67 | 21.98 | 45.69 | **18.67** | **7.35** | **3.80** | **2.26** | **16.89** | **20.24** |
| DKFPE | 33.67 | **16.27** | **3.98** | **24.28** | **50.46** | 16.74 | 6.18 | **2.87** | **1.55** | 15.72 | 18.69 |
| +S3-ESC | **34.27** | 16.54 | 3.77 | 23.63 | 49.71 | **17.18** | **6.25** | **2.87** | 1.50 | **16.03** | **19.07** |

sensitivity in Appendix D. Further implementation details are provided in Appendix B.

### 4.2 Main Results

Table 1 presents the main results. Applying S3-ESC consistently improves strategy prediction accuracy across all six baselines, with improvements ranging from +0.60pp (DKFPE) to +4.50pp (BlenderBot-Joint). This improvement across diverse architectures demonstrates the model-agnostic nature of our approach. In terms of response generation, S3-ESC improves or maintains BLEU and F1 scores for most baselines. BlenderBot+Ours achieves improvements across nearly all generation metrics, including BLEU-3 (3.63 → 3.88), BLEU-4 (2.05 → 2.27), and F1 (19.85 → 20.20). KEMI+Ours shows particularly notable gains in BLEU-1 (20.83 → 21.01), Dist-2 (12.64 → 13.46), and F1 (20.70 → 20.80). MISC+Ours improves ACC from 30.85 to 34.24 (+3.39pp) while also improving BLEU-4 (2.49 → 2.59) and ROUGE-L (17.45 → 17.63). Across most baselines, PPL is maintained or slightly improved. DKFPE+Ours improves ACC (33.67 → 34.27) along with BLEU-1 (16.74 → 17.18), ROUGE-L (15.72 → 16.03), and F1 (18.69 → 19.07), with only a slight PPL increase (16.27 → 16.54). PMPC+Ours presents a trade-off: while ACC improves substantially (32.10 → 35.13, +3.03pp) and BLEU-3/4 increase, Dist-1 slightly decreases (3.92 → 3.67). This suggests that as the model becomes more strategy-aware, it produces more contextually appropriate responses, reflected in higher BLEU and F1 scores, at the cost of slightly reduced lexical diversity.

Table 2: Ablation study on BlenderBot-Joint: Single-component (top) and pairwise (bottom) removal.

| Condition | ACC↑ | B-3↑ | B-4↑ | R-L↑ | F1↑ |
|---|---|---|---|---|---|
| Full | **35.77** | **3.88** | **2.27** | **16.91** | **20.20** |
| −SASRec | 33.10 | 3.53 | 1.99 | 16.29 | 19.54 |
| −SS | 34.49 | 3.79 | 2.26 | 16.55 | 19.74 |
| −Dec LoRA (DL) | 35.38 | 3.66 | 2.17 | 16.04 | 19.28 |
| −Enc LoRA | 33.92 | 3.71 | 2.13 | 16.35 | 19.62 |
| −LoRA All | 34.70 | 3.69 | 2.14 | 16.56 | 19.90 |
| −(DL + SS) | 35.27 | 3.86 | 2.33 | 16.10 | 19.35 |
| −(DL + SASRec) | 33.42 | 3.51 | 2.00 | 15.92 | 19.17 |
| −(SS + SASRec) | 33.38 | 3.67 | 2.15 | 16.50 | 19.82 |

### 4.3 Ablation Study

To assess the contribution of each component, we conduct ablation experiments (Table 2).

**Single-component ablation.** Removing our sequential strategy predictor (−SASRec) causes the largest ACC drop (35.77 → 33.10), confirming that the strategy sequence ensemble is the primary driver of strategy prediction improvement. Removing Scheduled Sampling (−SS) reduces ACC to 34.49 and lowers F1 from 20.20 to 19.74, indicating its role in both strategy prediction and generation quality. Removing the decoder LoRA experts (−Dec LoRA) while retaining the sequential predictor preserves most of the ACC gain (35.38), but generation metrics drop notably (F1: 20.20 → 19.28; ROUGE-L: 16.91 → 16.04), confirming that selective LoRA primarily contributes to generation quality rather than strategy prediction.

Removing only the encoder LoRA (−Enc LoRA) also reduces ACC (35.77 → 33.92), suggesting that the shared encoder LoRA improves the representations used for strategy prediction. Removing all LoRA modules (−LoRA All) decreases both ACC (35.77 → 34.70) and generation quality.

**Pairwise ablation.** Removing both Dec LoRA and SASRec (−(Dec LoRA + SASRec)) yields the largest overall degradation, with ACC dropping to 33.42 and F1 to 19.17, underscoring their complementary roles in strategy prediction and generation. Removing SS and SASRec (−(SS + SASRec)) produces the lowest ACC (33.38), suggesting that both components independently contribute to prediction accuracy. Notably, even when both Dec LoRA and SS are removed (−(Dec LoRA + SS)), ACC remains relatively high (35.27) due to the retained sequential predictor ensemble, but generation metrics decrease (F1: 20.20 → 19.35).

### 4.4 Strategy Sequence Analysis

To validate our claim that strategy sequences contain predictive sequential patterns, we compare SASRec's sequence-only predictions against baselines that use the full dialog context.

As shown in Table 3, SASRec achieves 33.38% accuracy despite having no access to the dialog text. Among the five baselines with a strategy predictor (GLHG has none), SASRec outperforms four and remains competitive with DKFPE (33.67%). This result demonstrates that the sequential ordering of past strategies alone is a strong signal for predicting the next strategy, supporting our motivation for incorporating a strategy sequence model.

Table 3: Strategy prediction accuracy (%).

| Model | ACC (%) | Input |
|---|---|---|
| BlenderBot-Joint | 31.27 | Full dialog |
| MISC | 30.85 | |
| PMPC | 32.10 | |
| KEMI | 32.85 | |
| DKFPE | 33.67 | |
| SASRec (Ours) | 33.38 | Strategy seq. only |

We further compare SASRec against simpler predictors that also use only the strategy labels (Table 4). The n-gram baselines condition on only the last one or two strategies and fall at least 4.21pp short of SASRec. A lightweight RNN, which sees the full history, reaches 32.27% and remains 1.11pp below. The sequential signal is therefore not reducible to local transition counts, and attending directly over the entire strategy history is more effective than compressing it into a single recurrent state.

Table 4: Comparison with simpler predictors.

| Predictor | ACC (%) |
|---|---|
| Majority | 20.21 |
| Bigram | 26.85 |
| Trigram | 29.17 |
| Lightweight RNN | 32.27 |
| SASRec (Ours) | 33.38 |

**Generalization to ExTES.** To assess whether this sequential signal generalizes beyond ESConv, we evaluate SASRec on ExTES (Zheng et al., 2024), an ESC dataset with a different strategy taxonomy. Using only the past strategy sequence, it reaches 50.0% next-strategy accuracy, well above both a random baseline (6.25%) and the test-split majority-class baseline (23.6%), indicating that sequential strategy patterns are predictive beyond ESConv. Further details are in Appendix F.

**Generalization to a decoder-only LLM.** We further test whether this mechanism transfers across architectures by applying its strategy prediction component to a decoder-only LLM (LLaMA) on ESConv. As a decoder-only model has no encoder, we necessarily use a decoder-side probe in place of the encoder probe. The sequential ensemble improves strategy prediction over the base model (34.0% → 35.3%), indicating that the core mechanism is not tied to the encoder-decoder structure. Details are in Appendix G.

### 4.5 Turn-wise Analysis

To examine where the sequential strategy predictor contributes, we measure strategy prediction accuracy by dialog position on BlenderBot-Joint, comparing the probe alone with the full ensemble.

As shown in Table 5, the ensemble is 1.69pp below the probe at turns 1–2, where almost no strategy history exists. From turn 3 onward the ensemble is higher at every position, by 2.84 to 4.62pp. The gain tracks the accumulated strategy history rather than arising from ensembling alone.

### 4.6 Human Evaluation

We complement the automatic metrics with a pairwise human evaluation on 200 test cases (100 each from BlenderBot-Joint and PMPC), comparing

Table 5: Strategy prediction accuracy (%) by dialog position on BlenderBot-Joint. Probe denotes S3-ESC without the sequential strategy predictor, and Ensemble denotes the full S3-ESC.

| **Turn** | $n$ | **Probe** | **Ensemble** | $\Delta$ |
|---|---|---|---|---|
| 1–2 | 296 | 55.41 | 53.72 | −1.69 |
| 3–4 | 390 | 31.28 | 35.90 | +4.62 |
| 5–6 | 390 | 25.38 | 28.72 | +3.33 |
| 7+ | 1,725 | 31.42 | 34.26 | +2.84 |
| All | 2,801 | 33.10 | 35.77 | +2.68 |

Table 6: Results of the pairwise human evaluation (%). ‡ / † represent $p < 0.05$ / 0.1 (two-sided sign test).

| **S3-ESC vs.** | **BlenderBot-Joint** | | | **PMPC** | | |
|---|---|---|---|---|---|---|
| | Win | Lose | Tie | Win | Lose | Tie |
| Fluency | **16**† | 7 | 77 | **20**‡ | 1 | 79 |
| Identification | **24**‡ | 9 | 67 | **23**‡ | 6 | 71 |
| Comforting | **41**‡ | 20 | 39 | **48**‡ | 17 | 35 |
| Suggestion | **19** | 12 | 69 | **28**‡ | 7 | 65 |
| Overall | **42** | 28 | 30 | **53**‡ | 22 | 25 |

both backbones with and without S3-ESC. Following prior work (Liu et al., 2021; Deng et al., 2023), three annotators independently judge each pair along the five ESConv dimensions (Fluency, Identification, Comforting, Suggestion, and Overall), marking whether the S3-ESC response is better, worse, or comparable to the base model. We aggregate by majority vote and test each (backbone, dimension) cell with a two-sided sign test over Win/Lose pairs (ties excluded). Further details are provided in Appendix H.

Table 6 reports the results. On PMPC, S3-ESC is significantly preferred on all five dimensions, most strongly on Overall (53 vs. 22) and Comforting (48 vs. 17). On BlenderBot-Joint, S3-ESC wins significantly on Comforting and Identification and marginally on Fluency; for Suggestion and Overall it is preferred more often than the baseline (Overall 42 vs. 28), though not significantly. No dimension shows a significant preference for the baseline. Inter-annotator agreement is fair-to-moderate (Fleiss' $\kappa = 0.38$).

Qualitative case studies for each component are provided in Appendix I.

## 5 Conclusion

In this paper, we identified three key limitations of existing ESC approaches: the failure to model sequential strategy transitions, a training-inference mismatch in strategy prediction, and the reliance on shared decoders that limit strategy-specific generation. To address these issues, we proposed S3-ESC, a simple, modular framework, model-agnostic across encoder-decoder ESC backbones. Experiments demonstrate that S3-ESC consistently improves strategy prediction while maintaining or improving response quality. These results highlight the importance of modeling sequential dynamics and enabling strategy-specialized generation for effective emotional support conversations.

## 6 Limitations

Our approach has several limitations. First, our framework still relies on the accuracy of strategy prediction; although the sequential predictor and scheduled sampling mitigate this dependence, incorrect predictions can still degrade generation. Second, our selective LoRA uses discrete routing to a single strategy-specific expert, which may be suboptimal when multiple strategies are relevant. Third, although designed to be lightweight, our components incur additional computational overhead from the ensemble prediction and multiple LoRA experts, which we quantify in Appendix C.

Fourth, our framework slightly increases PPL on DKFPE, and reduces lexical diversity on both DKFPE and PMPC (§4.2). Fifth, our human evaluation is conducted by graduate students who are proficient in English but not native speakers; we mitigate this with three annotators and clear evaluation guidelines. Sixth, our sequential predictor uses gold historical strategies during evaluation, following the standard ESC protocol (§4.1); this offline setting is more favorable than deployment, where past strategies would need to be predicted rather than observed, a property shared by every backbone compared in this work. Seventh, our ablation study (§4.3) isolates the contribution of each component only on BlenderBot-Joint, and whether the same effects hold for the other five backbones is not directly verified.

Finally, although we focus on encoder-decoder models on ESConv, we show that strategy prediction transfers to ExTES and to a decoder-only LLM; extending the complete framework to other datasets and architectures is left to future work.

## Ethics Statement

Our system is a research prototype and is not intended for deployment as a standalone source of mental health support. Like other ESC systems,

S3-ESC can produce inappropriate advice, premature suggestions, or false reassurance when the predicted strategy is incorrect or when the underlying backbone generates an unsafe response. We do not evaluate or claim any capability for detecting crisis situations (e.g., indications of self-harm) or for determining when a user should be referred to a human professional; such safety mechanisms are outside the scope of this work. Our experiments use ESConv, an English-language dataset collected from a specific user population; we do not evaluate our method on other languages or cultural contexts, and support strategies that are appropriate in one cultural setting may not transfer to another.

## Acknowledgments

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) under the Leading Generative AI Human Resources Development (IITP-2026-RS-2024-00360227) and the Artificial Intelligence Convergence Innovation Human Resources Development (IITP-2026-RS-2023-00254177) grants funded by the Korea government (MSIT).

## References

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Yang Deng, Wenxuan Zhang, Yifei Yuan, and Wai Lam. 2023. Knowledge-enhanced mixed-initiative dialogue system for emotional support conversations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4079–4095, Toronto, Canada. Association for Computational Linguistics.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. LoRAMoE: Alleviating world knowledge forgetting in large language models via MoE-style plugin. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1932–1945, Bangkok, Thailand. Association for Computational Linguistics.

Jiawang Hao and Fang Kong. 2025. Enhancing emotional support conversations: A framework for dynamic knowledge filtering and persona extraction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3193–3202, Abu Dhabi, UAE. Association for Computational Linguistics.

Clara E. Hill. 1999. *Helping Skills: Facilitating Exploration, Insight, and Action*. American Psychological Association.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Dongjin Kang, Sunghwan Kim, Taeyoon Kwon, Seungjun Moon, Hyunsouk Cho, Youngjae Yu, Dongha Lee, and Jinyoung Yeo. 2024. Can large language models be good emotional supporter? mitigating preference bias on emotional support conversation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15232–15261, Bangkok, Thailand. Association for Computational Linguistics.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *Proceedings of the IEEE International Conference on Data Mining*, pages 197–206.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Siyang Liu, Chujie Zheng, Orianna Demasi, Sahand Sabour, Yu Li, Zhou Yu, Yong Jiang, and Minlie Huang. 2021. Towards emotional support dialog systems. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3469–3483, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Wei Peng, Yue Hu, Luxi Xing, Yuqiang Xie, Yajing Sun, and Yunpeng Li. 2022. Control globally, understand locally: A global-to-local hierarchical graph network for emotional support conversation. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4324–4330. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Wei Peng, Ziyuan Qin, Yue Hu, Yuqiang Xie, and Yunpeng Li. 2023. Fado: Feedback-aware double controlling network for emotional support conversation. *Knowledge-Based Systems*, 264:110340.

Quan Tu, Yanran Li, Jianwei Cui, Bin Wang, Ji-Rong Wen, and Rui Yan. 2022. MISC: A mixed strategy-aware model integrating COMET for emotional support conversation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 308–319, Dublin, Ireland. Association for Computational Linguistics.

Xun Wu, Shaohan Huang, and Furu Wei. 2024. Mixture of LoRA experts. In *The Twelfth International Conference on Learning Representations*.

Yangyang Xu, Zhuoer Zhao, Xiao Sun, and Xun Yang. 2025. Prompt learning with multiperspective cues for emotional support conversation systems. *IEEE Transactions on Computational Social Systems*, 12(5):2648–2660.

Chao Zhang, Xin Shi, Xueqiao Zhang, Yifan Zhu, Yi Yang, and Yawei Luo. 2025. DecoupledESC: Enhancing emotional support generation via strategy-response decoupled preference optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 22189–22215, Suzhou, China. Association for Computational Linguistics.

Weixiang Zhao, Xingyu Sui, Xinyang Han, Yang Deng, Yulin Hu, Jiahe Guo, Libo Qin, Qianyun Du, Shijin Wang, Yanyan Zhao, Bing Qin, and Ting Liu. 2025. Chain of strategy optimization makes large language models better emotional supporter. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 15361–15381, Suzhou, China. Association for Computational Linguistics.

Weixiang Zhao, Yanyan Zhao, Shilong Wang, and Bing Qin. 2023. TransESC: Smoothing emotional support conversation via turn-level state transition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6725–6739, Toronto, Canada. Association for Computational Linguistics.

Zhonghua Zheng, Lizi Liao, Yang Deng, Libo Qin, and Liqiang Nie. 2024. Self-chats from large language models make small emotional support chatbot better. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11325–11345, Bangkok, Thailand. Association for Computational Linguistics.

## A Definitions of Strategies

The strategy definitions are adapted from Liu et al. (2021), grounded in Hill's Helping Skills Theory (Hill, 1999).

- **Question (Qu):** Asking for information about the problem to help the help-seeker articulate the issues they face.
- **Restatement or Paraphrasing (RP):** A concise rephrasing of the help-seeker's statements to help them see their situation more clearly.
- **Reflection of Feelings (RF):** Articulating and describing the help-seeker's feelings.
- **Self-disclosure (Sd):** Sharing similar experiences or emotions to express empathy with the help-seeker.
- **Affirmation and Reassurance (AR):** Affirming the help-seeker's strengths and capabilities, and offering reassurance and encouragement.
- **Providing Suggestions (PS):** Offering suggestions for change while being careful not to overstep.
- **Information (In):** Providing useful information, such as data, facts, opinions, or resources.
- **Others (Ot):** Exchanging pleasantries or using strategies outside the categories above.

## B Implementation Details

**Selective LoRA.** Selective LoRA uses rank $r = 8$, scaling factor $\alpha = 16$, and dropout $0.05$.

**Linear probe.** The probe is trained with a separate Adam optimizer (Kingma and Ba, 2015) (learning rate $1 \times 10^{-3}$). It is first trained jointly with the backbone, then re-trained for 100 epochs (batch size 256) on cached hidden states from the trained backbone, selected by validation accuracy.

**Sequential strategy predictor.** The SASRec predictor uses a hidden dimension of 256, 4 layers, 4 attention heads, and a maximum sequence length of 50. It is trained for 50 epochs (batch size 128, learning rate $5 \times 10^{-4}$).

**Ensemble.** The ensemble weight $\lambda$ is selected via grid search (step $0.05$) on the validation set. Backbone checkpoints are selected by validation perplexity, probe checkpoints by validation accuracy.

## C Efficiency Analysis

The overhead of S3-ESC is small at inference time. The selective LoRA experts add only rank-8 low-rank updates that merge into the existing projection matrices, leaving the decoding cost essentially unchanged, and the sequential ensemble adds about 1 ms per dialog. In terms of parameters, S3-ESC adds 11.44M in total, an identical absolute overhead across all backbones since they share the same generator and the SASRec module and probe are backbone-agnostic. Relative to the smallest backbone (87.5M BlenderBot-small), this is a $+13.1\%$ increase, and proportionally smaller for larger backbones.

## D Hyperparameter Study

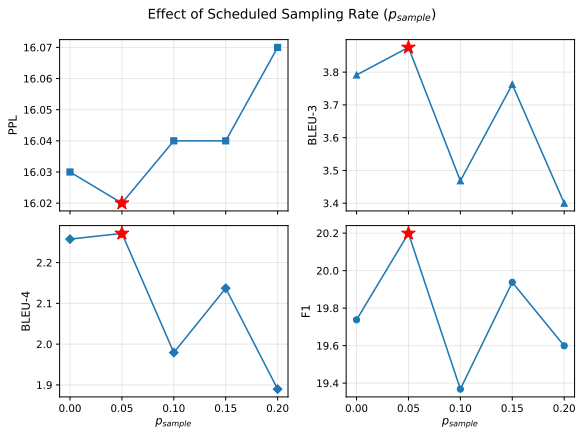



Figure 4: Hyperparameter Study

We investigate the effect of the scheduled sampling probability $p_s$ on generation quality, using BlenderBot as the backbone with selective LoRA and the sequential prediction ensemble applied. Fig. 4 presents the results across five values of $p_s$.

A small amount of scheduled sampling ($p_s = 0.05$) yields the best performance across PPL (16.02), BLEU-3 (3.88), BLEU-4 (2.27), and F1 (20.20), all improving over the teacher-forcing baseline ($p_s = 0.0$). However, higher sampling probabilities ($p_s \geq 0.10$) degrade performance, with BLEU-4 dropping to 1.98 at $p_s = 0.10$ and 1.89 at $p_s = 0.20$. This suggests that while a small amount of exposure to predicted strategies during training reduces the mismatch, excessive noise destabilizes the generation process. The sensitivity to $p_s$ is consistent with findings in scheduled sampling for sequence generation (Bengio et al.,

2015), where moderate perturbation is beneficial but excessive perturbation harms convergence.

## E Probing Position Analysis

We compare two probing positions for strategy prediction: an encoder-side probe using attention-masked mean pooling of encoder hidden states, and a decoder-side probe using the BOS (beginning-of-sequence) token representation.

As shown in Table 7, the encoder probe consistently outperforms the decoder probe across all $p_s$ values, with the gap ranging from $+1.1$pp ($p_s = 0.05$) to $+2.8$pp ($p_s = 0.10$), suggesting that the encoder's aggregated context representation provides a more informative signal for strategy prediction than the decoder's initial hidden state. Moreover, the decoder probe's accuracy decreases monotonically as $p_s$ increases ($31.85 \rightarrow 30.77$), while the encoder probe does not exhibit such degradation, indicating that the encoder representations are more robust to the perturbation introduced by scheduled sampling. Based on these findings, we adopt the encoder probe as the default strategy predictor for all experiments.

Table 7: Strategy prediction accuracy (%) by probing position across $p_s$ values, on BlenderBot.

| $p_s$ | **Encoder** | **Decoder** | $\Delta$ |
|---|---|---|---|
| 0.00 | 33.56 | 31.85 | +1.7 |
| 0.05 | 32.63 | 31.56 | +1.1 |
| 0.10 | **34.31** | 31.52 | +2.8 |
| 0.15 | 33.35 | 31.17 | +2.2 |
| 0.20 | 33.35 | 30.77 | +2.6 |

## F Sequential Strategy Prediction on ExTES

To test whether the predictiveness of strategy sequences is specific to ESConv or holds more generally, we evaluate the SASRec sequential predictor (§3.6) on ExTES (Zheng et al., 2024), a large-scale ESC dataset synthesized by prompting an LLM. ExTES uses a different taxonomy of 16 support strategies, distinct from the 8 strategies in ESConv.

**Data preparation.** ExTES provides no official data split, so we split it at the conversation level with a 70:15:15 ratio (seed 42), following our ESConv protocol; splitting by conversation prevents turns from the same dialog leaking across splits. The strategy labels in ExTES are noisy: of all labeled turns, the 16 canonical strategies account for 98.1%, while the remaining 1.9% are typographical or non-standard variants. We apply a strict normalization that matches only the 16 canonical strategies (case- and whitespace-insensitive) and drop the unmatched 1.9% to avoid subjective mapping. After preprocessing, the test split contains 11,699 strategy turns.

**Results.** Using only the past strategy sequence, the SASRec predictor reaches 50.0% top-1 accuracy and 0.405 macro-F1 over the 16 strategies on the test split. For reference, a random baseline achieves 6.25% and the test-split majority-class baseline (Emotional Validation, the most frequent strategy) achieves 23.6%; the predictor exceeds both by $8.0\times$ and $2.1\times$, respectively. This confirms that sequential strategy patterns are predictive beyond ESConv, indicating that the signal is not specific to a single dataset.

## G Decoder-only LLM Experiment

To test whether the strategy prediction mechanism of S3-ESC extends beyond encoder-decoder backbones, we apply its sequential ensemble to a decoder-only LLM on ESConv. Since a decoder-only model has no encoder, we place the strategy probe on the final input token's last-layer representation. In a causal decoder-only model, the BOS token attends only to itself and carries no dialog context, whereas the final input token attends to the entire context; it therefore serves as the analog of the decoder BOS that, in the encoder-decoder setting, accesses the context through cross-attention (§3.4, Appendix E).

**Setup.** We use LLaMA-3.1-8B-Instruct as the backbone, fine-tuned with standard 4-bit QLoRA (rank 8, $\alpha = 16$, dropout 0.05) and the base weights frozen. We train with a learning rate of $1 \times 10^{-4}$ (cosine schedule, warmup ratio 0.03), an effective batch size of 16, and a maximum sequence length of 1024. During training, the first epoch uses ground-truth strategy warmup and the second applies scheduled sampling. The sequential predictor (SASRec) reuses the same model trained on ESConv strategy sequences (§3.6) without retraining, underscoring its model-agnostic property.

**Strategy prediction.** As shown in Table 8, ensembling the decoder-side probe with the sequential strategy model improves strategy prediction accuracy to 35.3%, above the lr-matched base model

(34.0%). This shows that our sequential ensemble, the core strategy prediction mechanism of S3-ESC, transfers to a decoder-only setting as well, consistent with our findings on encoder-decoder backbones (§4.4).

Table 8: Strategy prediction accuracy (%) on the decoder-only LLM backbone. The base model and our method use the same learning rate.

| Condition | ACC |
|---|---|
| Base model | 34.0 |
| + Sequential ensemble | **35.3** |

## H Human Evaluation Details

**Dimensions.** Following Liu et al. (2021), annotators judge each pair along five ESConv dimensions: **Fluency** (which response is more fluent and understandable), **Identification** (which explores and identifies the seeker's problem more effectively), **Comforting** (which is more skillful in comforting the seeker), **Suggestion** (which gives more helpful suggestions), and **Overall** (which provides more effective emotional support).

**Setup.** Evaluation is fully blind: the two responses are shown left and right under a fixed seeded assignment, and the ground-truth response and strategy labels are withheld. Following Zhao et al. (2023), the pre-conversation situation is also withheld, since a supporter can only understand the seeker's circumstances as the dialog unfolds.

**Annotators.** The three annotators are graduate students who participated voluntarily and were given a small compensation for their time.

## I Case Study

We present qualitative examples to illustrate the behavioral differences between the baseline and S3-ESC. Table 9 shows three representative cases from the BlenderBot backbone, each illustrating the effect of one S3-ESC component.

In Case A, the user has just stated a clear preference ("I definitely prefer in-school studies"), and the ground-truth strategy is *Question*. The joint probe predicts *Affirmation and Reassurance*, the same incorrect strategy as the baseline, whose response ("i am glad you are doing both ways") is a generic affirmation that misreads the user's stated preference. The SASRec sequence branch, however, predicts *Question*, and the ensemble of the two selects *Question*; conditioned on it, +S3-ESC asks a relevant follow-up question. This illustrates the effect of the SASRec sequence ensemble: the sequential strategy signal overrides the probe's incorrect prediction and recovers the ground-truth *Question* that the baseline missed.

In Case B, both models correctly predict *Providing Suggestions* for a user worried that helping their mother might be taken as criticism. The baseline fails to realize this strategy: it only speculates about the mother's feelings ("maybe she's just feeling a little down and worried") and offers no course of action. In contrast, +S3-ESC produces a concrete, hedged suggestion that directly addresses the user's worry, advising the user to talk to her and reassure her that the help is not a judgment, aligning with the ground-truth intent. This illustrates the benefit of selective LoRA: the strategy-specific expert steers decoding toward content that genuinely realizes *Providing Suggestions*, even when the baseline, despite predicting the same strategy, fails to produce an actionable suggestion.

In Case C, both models miss the ground-truth *Providing Suggestions* and instead predict *Affirmation and Reassurance*. Because the predicted strategy conditions generation as the first decoder token, an incorrect strategy can derail the response: the baseline, never trained under such mismatch, degenerates into repetition, echoing "it sounds like you are doing all you can" twice, whereas scheduled sampling exposes +S3-ESC to incorrect strategies during training and yields a fluent, grounded response ("you sound like a very strong person and i am sure your son will feel the same way") even when the predicted strategy is off.

Table 9: Case study on BlenderBot. Each case illustrates the effect of one S3-ESC component: the SASRec sequence ensemble (Case A), selective LoRA (Case B), and scheduled sampling (Case C). GT: ground-truth.

| | |
|---|---|
| **Case A** (idx=58): The baseline and the joint probe both predict an incorrect strategy, while the SASRec sequence ensemble recovers the ground-truth *Question*.<br>*Situation*: The user feels anxious about impending academic deadlines. | |
| Context | **Supporter** [*Question*]: are you taking a lot of courses?<br>**User**: 2 courses. but I am pretty much using full time capacity on them. devoting full-time I mean.<br>**Supporter** [*Reflection of Feelings*]: do you like doing school online better than in person?<br>**User**: Have to finish one at the end of this month and end of next month for the other one.<br>**Supporter** [*Affirmation and Reassurance*]: Thats good the more time you have to devote the better!<br>**User**: I have done both methods. I definitely prefer in-school studies. |
| GT | **Strategy**: *Question*<br>Do you feel it is more stressful online? |
| Baseline | **Strategy**: *Affirmation and Reassurance*<br>i am glad you are doing both ways. it sounds like you are doing your best. |
| +Ours | **Strategy**: *Question*<br>are you doing online schooling? or are you just doing the online classes? |
| **Case B** (idx=504): Both models correctly predict *Providing Suggestions*, but +S3-ESC realizes the strategy while the baseline fails to offer any suggestion.<br>*Situation*: The user fears that helping their mother will be taken as criticism. | |
| Context | **Supporter** [*Self-disclosure*]: I have had personal disputes with my parents in the past. Most of the time there's other things going on that makes a person act different.<br>**User**: Maybe I could just take deep breath and kind of step back a minute before I respond.<br>**Supporter** [*Providing Suggestions*]: It would be good to do that. Maybe try to have a conversation and tell her you support her no matter what she's going through.<br>**User**: I guess the best time to do that was when she is in a very good mood.<br>**Supporter** [*Providing Suggestions*]: Do some chores or something that helps her out and just let her know you want to have a serious talk some time.<br>**User**: But what if she thinks me helping her out is criticizing the way she does things? That seems like the way she would be. |
| GT | **Strategy**: *Providing Suggestions*<br>Try giving reassurance that you want to show you care. |
| Baseline | **Strategy**: *Providing Suggestions*<br>maybe she's just feeling a little down and worried. maybe she's just feeling a little down and anxious about this whole situation. |
| +Ours | **Strategy**: *Providing Suggestions*<br>maybe it would be a good time to talk to her about it. just let her know that you're here to help and she's not going to judge you. |
| **Case C** (idx=1831): Both models miss the ground-truth *Providing Suggestions* and predict *Affirmation and Reassurance*, but +S3-ESC stays fluent while the baseline degenerates into repetition.<br>*Situation*: An elderly user feels unable to enjoy retirement while their adult son is struggling financially. | |
| Context | **User**: Thank you so much for your support. It is tough and there is nothing to do.<br>**Supporter** [*Question*]: Do you have family you can talk to?<br>**Supporter** [*Providing Suggestions*]: Have you considered volunteering? I ask because my neighbor bakes for the local food bank and it helps her.<br>**User**: Yes, I do, but I can't talk to my son in Florida because he lost his job and havng tough time. I don't feel like I should enjoy my life when he is suffering. We may need to help him financially. I use to volunteer at a nursing home, but I am afraid to now with the virus<br>**Supporter** [*Affirmation and Reassurance*]: That is reasonable. And it sounds like you are doing all you can. |
| GT | **Strategy**: *Providing Suggestions*<br>Could you teach from home online and keep yourself safe? Or perhaps provide your son financial support? |
| Baseline | **Strategy**: *Affirmation and Reassurance*<br>that is understandable. and it sounds like you are doing all you can. and it sounds like you are doing all you can. |
| +Ours | **Strategy**: *Affirmation and Reassurance*<br>you sound like a very strong person and i am sure your son will feel the same way. |